

RS03-L*
RS04-A* RH11

RH11-RS03/RS04 BFT
CZRSBH0

AH-9315H-MC
FICHE 1 OF 1

FEB 1981
COPYRIGHT © 73-80
MADE IN USA



1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33

.REM @

IDENTIFICATION

PRODUCT CODE: AC-9314H-MC
PRODUCT NAME: CZRSBH0 RH11-RS03/RS04 BFT
PRODUCT DATE: OCT 1980
MAINTAINER: DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973,1980 BY DIGITAL EQUIPMENT CORPORATION

CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.3	PRELIMINARY PROGRAMS
3.	LOADING PROCEDURE
4.	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	PROGRAM AND/OR OPERATING PROCEDURE
5	OPERATIONAL SWITCH SETTINGS
5.2	SUBROUTINE ABSTRACT
6.	ERRORS
7.	RESTRICTIONS
8.	MISCELLANEOUS
8.1	EXECUTION TIME
8.2	STACK POINTER
9.	WRITE LOCK TEST
10.	TEST DESCRIPTION

34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76

1. ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST RS03,RS03/LA AND RS04 DRIVES.

THIS IS A BASIC FUNCTION DIAGNOSTIC WHICH IS USED TO VERIFY THAT THE (RH11) CONTROLLER AND THE (RS03,RS03/LA OR RS04) DISKS ARE OPERATING CORRECTLY. THIS IS NOT A RELIABILITY DIAGNOSTIC AND THEREFORE SHOULD NOT BE USED AS ONE. THIS PROGRAM CAN TEST UP TO 8 DRIVES. THE DRIVES CAN BE INTERMIXED AND IN ANY ORDER.

IF THE OPERATOR WOULD LIKE TO CHECK THE DISK REGISTERS PRIOR TO ENTERING THIS DIAGNOSTIC, THERE ARE SOME ROUTINES IN THE BACK OF THE DIAGNOSTIC WHICH CAN BE USED. THESE ROUTINES WILL ALLOW THE OPERATOR TO LOAD THE REGISTERS THROUGH THE SWITCHES. PLEASE REFERENCE THE STARTING ADDRESSES THAT WILL TEST THE REGISTERS YOU DESIRE.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP11 STANDARD COMPUTER WITH A MINIMUM OK 8K OF MEMORY, AND AN RH11 CONTROLLER WITH A RS03, RS03/LA OR RS04 DISK.

2.3 PRELIMINARY PROGRAMS

NONE

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR ABS TAPES.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SEE 5.1.1 (ALL DOWN FOR WORST CASE TESTING)

4.2 STARTING ADDRESS

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY USING ABS LOADER.

1. STARTING ADDRESS 200.

- A. SET SWITCHES (SEE SEC 5.1.1) ALL DOWN FOR WORST CASE. IF SWITCHLESS CPU, SIMPLY
- B. PRESS START.
- C. THE PROGRAM WILL LOOP AND BELL WILL RING ONCE EVERY PASS
- D. THE DISPLAY ON THE 11/45 WILL SHOW THE ITERATION COUNT IN THE LEFT BYTE AND TEST NUMBER IN THE RIGHT. TO USE, SET THE DATA DISPLAY SWITCH TO THE DISPLAY POSITION.
- E. THE PROGRAM WILL TEST ALL RS03, RS03/LA AND RS04 DISKS.

2. STARTING ADDRESSES FOR TESTING THE RH11-RS03/LA/04 REGISTERS USING THE SWITCH REGISTER. ON SWITCH-LESS MACHINES THESE ROUTINES ARE USEFUL FOR SCOPING. SIMPLY STRIKE ^G ANYTIME AFTER PRESSING START TO ENTER OR CHANGE VALUE DESIRED.

- | | |
|--------|-----------------------------|
| A. 250 | WORD COUNT REGISTER TEST |
| B. 254 | BUS ADDRESS REG. TEST |
| C. 260 | DISK ADDRESS REG. TEST |
| D. 264 | DRIVE STATUS REG. TEST |
| E. 270 | ERROR REG. TEST |
| F. 274 | LOOK AHEAD REG. TEST |
| G. 300 | RSCS2 REG. TEST |
| H. 304 | ATTENTION SUMMARY REG. TEST |
| I. 310 | MAINTENANCE REG. TEST |
| J. 314 | RSCS1 REG TEST |

5. OPERATIONAL SWITCH SETTINGS

THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC.176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(I.E.) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

- 1. <CR> IF NO CHANGES ARE TO BE MADE.
- 2. 6 DIGITS 0-7 TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE; LAST DIGIT FOLLOWED BY <CR>.
- 3. ^U TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ^G (CNTRL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

SWITCH SETTINGS ARE:

SW<15> = 1 HALT ON ERROR
SW<14> = 1 LOOP ON TEST
SW<13> = 1 INHIBIT TYPEOUTS
SW<12> = 1 INHIBIT OBUFSV FROM CHANGING WHEN LOOKING
FOR MEMORY ON -B- PORT
SW<11> = 1 INHIBIT ITERATIONS OF SUBTEST
SW<10> = 1 BELL ON ERROR
0 BELL ON PASS COMPLETE
SW<09> = 1 LOOP ON ERROR
SW<08> = 1 LOOP ON TEST IN SW<7:0>

5.2 SUBROUTINE ABSTRACTS

5.2.1 SCOPE

THIS SUBROUTINE CALL IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT RECORDS THE STARTING ADDRESS OF EACH SUBTEST AS IT IS BEING ENTERED IN LOCATION 'LAD'. IF A SCOPE LOOP IS REQUESTED, THE CURRENT SUBTEST WILL BE LOOPED UPON. SW<11> ON A 1 INHIBITS ITERATION OF SUBTESTS. THE CONTENTS OF LAD MAY BE USED TO DETERMINE THE LAST SUBTEST SUCCESSFULLY COMPLETED.

5.2.2 HLT

THIS ROUTINE PRINTS OUT AN ERROR MESSAGE (SEE 6.1). TO INHIBIT TYPEOUTS, PUT SW<13> ON A 1.

5.2.3 TRAPCATCHER

A ".+2" - "HALT" SEQUENCE IS REPEATED FROM 0 - 776 TO CATCH ANY UNEXPECTED TRAPS. THUS ANY UNEXPECTED TRAPS OR INTERRUPTS WILL HALT AT THE VECTOR + 2.

6. ERRORS

6.1 ERROR PRINTOUT

THE FORMAT IS AS FOLLOWS:

ADR CS1 = ----- CS2 = ----- ER = -----
GOOD = ----- BAD = -----

WHERE:

CS1,CS2,ER ETC. = RS11 DISK REGISTERS.
GOOD = EXPECTED DATA.
BAD = DATA RECEIVED.

TO FIND THE FAILING TEST, LOOK AT THE LISTING ABOVE THE ADDRESS TYPED.

6.2 ERROR RECOVERY

RESTART AT 200

7. RESTRICTIONS

NONE

8. MISCELLANEOUS

8.1 EXECUTION TIME

A BELL WILL RING WITHIN 1 MINUTE WITH ALL SWITCHES DOWN.

8.2 STACK POINTER

STACK IS INITALLY SET TO 500

9. WRITE LOCK TEST

THE WRITE LOCK TEST REQUIRES OPERATOR INTERVENTION. THE STARTING ADDRESS FOR THIS TEST IS 220. THE PROGRAM WILL TELL THE OPERATOR WHICH SWITCHES HAVE TO BE SET.

10. TEST DESCRIPTION

296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351

1. TEST RSCS2

CLEAR ALL READ/WRITE BITS AND CHECK. SET ALL R/W BITS AND CHECK. NOW CLEAR AND RECHECK.

2. TEST FOR ONLINE DRIVES

SET ERROR BITS IN RSER. THIS CAUSES ATTENTION SUMMARY BITS TO SET IN RSAS. DO FOR ALL DRIVES. RSAS HAS NOT YET BEEN TESTED. SO IN THE CASE OF NO BITS IN RSAS SETTING, DRIVE 0 IS TESTED.

3. RESET TEST FOR REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB, AND RSMR. DO A RESET AND TEST ALL R/W BITS TO BE CLEARED.

4. SET AND CLEAR ALL REGISTERS

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB AND RSMR AND TEST. SET ALTERNATE BITS AND CHECK TO MAKE SURE BITS ARE NOT TIED TOGETHER. NOW SET ALL BITS AND CLEAR THEM TO MAKE SURE ALL CAN BE CLEARED ONCE SET.

5. RANDOM NUMBER TEST FOR RSWC AND RSDA

THIS TEST GENERATES RANDOM NUMBERS AND LOADS THEM INTO RSWC, RSDA AND RSBA.

6. TEST "CLEAR BIT" IN RSCS2

SET ALL R/W BITS IN RSCS1, RSCS2, RSBA, RSDA, RSER, RSWC, RSDB, AND RSMR. SET CLEAR BIT IN RSCS2. NOW TEST ALL R/W BITS FOR 0 IN ALL THE ABOVE REGISTERS.

7. TEST DLT AND TRE BITS

DO A READ FROM THE SILO. THIS SHOULD CAUSE A DLT AND A TRE ERROR BECAUSE THE SILO IS EMPTY.

8. CLEAR DLT AND TRE

CLEAR BY SETTING TRE IN RSCS1 AND TEST.

9. LOAD RSDB WITH ALL ONES AND ALL ZEROS

LOAD RSDB WITH A WORD OF ZEROS AND A WORD OF ONES. WAIT FOR "OR" TO SET AND THEN CHECK OUTPUT OF SILO. IF OR DID NOT SET ERROR MESSAGE APPEARS.

10. TEST FOR 66 LOCATIONS IN SILO

THIS IS DONE BY PUTTING A BINARY COUNT IN EVERY LOCATION AND

352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
398
399
400
401
402
403
404
405
406
407

CHECKING THE OUTPUT FOR 66 WORDS.

11. TEST DLT ERROR

THIS IS DONE BY LOADING THE SILO WITH 67 WORDS WITHOUT READING ANY OUT. THIS SHOULD CAUSE DLT TO SET.

12. FLOAT A '1' AND A '0' THROUGH THE SILO

LOAD THE SILO WITH A WORD OF ZEROS AND FLOAT A '1' THROUGH THE WORD. THEN LOAD THE SILO WITH A WORD OF ALL ONES AND FLOAT A '0' THROUGH THE WORD. CHECK THE OUTPUT OF THE SILO FOR THE CORRECT ANSWER.

13. TEST NO-OP FUNCTION

THE NO-OP FUNCTION IS TESTED WITH AND WITHOUT ERROR BITS SET. ALL THE REGISTERS ARE CHECKED AFTER BOTH CASES.

14. TEST DRIVE CLEAR FUNCTION

FIRST SET ALL R/W BITS IN RSDA, RSWC, RSER, AND RSMR. DO A DRIVE CLEAR FUNCTION. NOW TEST ALL REGISTERS FOR CORRECT DATA.

15. EXECUTE A ONE WORD WRITE FUNCTION

SET RSWC TO -1. MOV -1 INTO OUTBUF. LOAD RSBA WITH OUTBUF. DO A WRITE TEST RDY BIT FOR 0 THEN WAIT FOR IT TO SET. TIME OUT TO ERROR IF RDY BIT DOESN'T SET AND CHECK FOR ERROR CONDITIONS. TEST RSDA FOR CORRECT ADDRESS. TEST WORD COUNT FOR 0. THIS IS TESTED ON -A- AND -B- PORT.

16. EXECUTE A ONE WORD WRITE CHECK

SET UP RSDA, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION TEST. DO A WRITE CHECK FUNCTION. TEST RDY AS DONE IN THE WRITE TEST. CHECK FOR WRITE CHECK ERROR. THEN TEST RSDA, RSWC AND RSBA FOR CORRECT DATA. THIS IS TESTED ON -A- AND -B- PORT.

17. TEST READ FUNCTION

SETUP RSDA, RSBA, RSWC AND OUTBUF AS IN THE WRITE FUNCTION DO A READ FUNCTION. TEST RDY BIT AS DONE IN THE WRITE FUNCTION. TEST FOR ERRORS. ALSO TEST RSDA, RSWC AND RSBA FOR CORRECT DATA. THIS IS TESTED ON -A- AND -B- PORT.

18. TEST BLOCK SEARCH FUNCTION, PIP AND DRY BITS AND ADDR. CONF. BIT

DO A BLOCK SEARCH FOR SECTOR 32, LOOP ON ADDR. CONF. BIT IN RSMR. IF IT DOESN'T SET, TIMEOUT. WHEN YOU GET THERE DO A BLOCK SEARCH FOR SECTOR 0. NOW WE KNOW THAT WE HAVE TIME TO

408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463

TEST FOR DRY AND PIP BITS BEFORE FINDING SECTOR 0. FOR PIP SHOULD SET AND DRY SHOULD CLEAR BEFORE FINDING SECTOR 0. ONCE SECTOR 0 IS FOUND PIP SHOULD CLEAR AND DRY SHOULD SET. IF DRY DOES NOT SET A TIME OUT ERROR WILL OCCUR INDICATING SECTOR 0 WAS NOT FOUND. SC IN RSCS1 SHOULD ALSO SET. RSBA AND RSWC SHOULD NOT MOVE, THIS IS ALSO TESTED.

19. ILLEGAL FUNCTION CODE TEST

IN THIS TEST RSBA, RSWC AND RSDA ARE SET UP AS IF TO DO A LEGAL FUNCTION. AN ILLEGAL FUNCTION IS THEN EXECUTED. THE PROGRAM TEST FOR ILF AND ERR BITS TO SET. RSBA, RSWC AND RSDA ARE ALSO TESTED FOR CORRECT DATA. THIS IS DONE FOR ALL THE ILLEGAL FUNCTIONS.

FOR AN AID IN TROUBLE SHOOTING THE ILLEGAL FUNCTION CODE CAN BE LOADED INTO LOCATION ILLTAB OR ILFTB2, DEPENDING ON WHICH ILLEGAL FUNCTION TEST YOU WISH TO LOOP ON. IN THE NEXT LOCATION, FOLLOWING THE ILLEGAL FUNCTION, A 0 MUST BE LOADED. NOW BY SETTING SWITCH 14 (LOOP ON TEST), YOU WILL LOOP ON THE ILLEGAL FUNCTION.

20. TEST PAR IN RSER

SET PAR IN RSER AND CHECK. ALSO TEST ERR IN RSDS TO SET BECAUSE OF THE PAR SETTING.

21. TEST DPR AND MOL IN RSDS

BOTH THESE BITS SHOULD BE SET IN RSDS IF THE DRIVE IS ON LINE AND UP TO SPEED.

22. LOOK AHEAD TEST

FIRST CHECK TO SEE IF SECTOR FRACTION BITS ARE MOVING. NOW SET RSDA TO 0 AND INCREMENT IT EVERY TIME THE ADDR.CONF BIT SETS. IF THE ADDR.CONF BIT DOES NOT SET IN A CERTAIN LENGTH OF TIME, A TIME OUT ERROR OCCURS.

23. PARITY TEST

THIS WILL TEST THE PARITY LOGIC ONLY IF THERE IS PARITY MEMORY ON THE SYSTEM IN LESS THAN 28K. IT WILL WRITE BAD PARITY IN A MEMORY LOCATION THEN TRY TO DO A WRITE TO THE DRIVE FROM THAT LOCATION. THIS SHOULD CAUSE A PARITY ERROR.

24. TEST WRITE CHECK ERROR

IN THIS TEST THE PROGRAM WRITES A -1 ON TO THE DISK. A 0 IS NOW FLOATED THROUGH THE WORD IN THE BUS ADDRESS LOCATION, AND A WRITE CHECK FUNCTION IS DONE. THE WCE BIT IN RSCS2 SHOULD SET AND SHOULD CAUSE THE TRE BIT IN RSCS1 TO SET. THESE BITS ARE THEN CLEARED. A WORD OF 0 IS NOW WRITTEN ON THE DISK AND A 1 IS FLOATED THROUGH THE WORD IN THE BUS ADDRESS AND THE

464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519

WRITE CHECK FUNCTION TEST IS REPEATED.

25. TEST PROGRAM ERROR BIT IN RSCS2

HERE THE PROGRAM ATTEMPTS TO INITIATE A DATA TRANSFER OPERATION WHILE THE CONTROL IS CURRENTLY PERFORMING ONE. THIS SHOULD CAUSE PGE TO SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED. RSWC IS ALSO TESTED FOR IT SHOULD NOT BE 0 FOR THE CURRENT OPERATION SHOULD HAVE BEEN ABORTED DUE TO THE PGE ERROR.

26. TEST RMR IN RSER

HERE A WRITE COMMAND IS GIVEN AND DURING ITS EXECUTION THE PROGRAM TRYS TO MODIFY THE RSDA REG. THIS SHOULD CAUSE THE RMR BIT TO SET WHICH CAUSES THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

27. TEST DCK IN RSER

HERE A WRITE COMMAND IS GIVEN THEN DURING THIS FUNCTION A DRIVE CLEAR COMMAND IS GIVEN. THIS SHOULD CAUSE THE DCK BIT TO SET WHICH SHOULD CAUSE THE ERR BIT TO SET. THESE BITS ARE THEN CLEARED.

28. TEST DISK ADDRESS REGISTER

LOAD THE LAST DISK ADDRESS (7777) INTO RSDA. DO A ONE WORD WRITE AND CHECK THAT RSDA INCREMENTED TO 10000.

29. TEST IAE ERROR

DO A ONE WORD WRITE BUT FIRST SET RSDA TO AN INVALID ADDRESS SUCH AS 10000. THIS SHOULD CAUSE A IAE ERROR WHICH WILL CAUSE ERR, ATA AND SC BITS TO SET. THESE BITS ARE THEN CLEARED BY LOADING A 1 INTO ATA IN RSAS.

30. TEST FOR NON-EXISTENT DISK ERROR

FIRST FIND A DRIVE THAT IS NOT ON THE SYSTEM OR OFF LINE. NOW TRY TO DO A ONE WORD WRITE TO THAT DRIVE. NEP IN RSCS2 SHOULD SET WHICH SHOULD CAUSE TRE TO SET. THESE BITS ARE THEN CLEARED BY MOVING A 1 INTO TRE.

31. TEST DAO IN RSER AND LBT IN RSDS

SET RSDA TO ITS LAST ADDRESS. NOW WRITE ONE SECTOR PLUS ONE WORD. DAO SHOULD SET AND LBT SHOULD SET. THESE SHOULD CAUSE ERR, ATA, TRE AND SC TO SET. THESE ARE CLEARED BY DOING A CLEAR.

32. TEST BAI IN RSCS2

SET BAI IN RSCS2. DO A ONE WORD WRITE AND CHECK RSBA TO SEE

IF IT INCREMENTED.

33. TEST NON-EXISTENT MEMORY ERROR BIT IN RSCS2

SET BITS A16 AND A17 IN RSCS1 FOR AN 18 BIT ADDRESS. MOV
173000 INTO RSBA. MOV -1000 INTO RSWC AND DO A WRITE
FUNCTION. THE NEM BIT SHOULD SET AND SHOULD CAUSE TRE TO
SET. CLEAR THESE BITS BY LOADING A 1 INTO TRE.

34. TEST FOR ZERO'S IN A PARTIALLY FILLED SECTOR

FIRST WRITE A COMPLETE SECTOR WITH ALL ONES. THEN DO A ONE
WORD WRITE. THE REMAINING 63 WORDS SHOULD BE WRITTEN AS
ZERO'S. NOW DO A WRITE CHECK TO COMPARE FOR THESE ZERO'S.

35. PRIORITY INTERRUPT TEST

HERE THE PROGRAM ENABLES THE INTERRUPT AND DOES A ONE WORD
WRITE FUNCTION. THE PROGRAM SHOULD NOT TRAP UNTIL THE
PROCESSOR IS DROPPED TO PRIORITY 4.

36. DYNAMIC FUNCTION TEST

WHILE ONE DRIVE IS READING, THE UNIT # IN RSCS2 IS MODIFIED.
IF THERE IS ANOTHER DRIVE ON THE SYSTEM, A DRIVE SEARCH IS
PERFORMED ON IT. THIS IS ALL DONE WHILE THE FIRST DRIVE IS
STILL READING.

ECO SECTION

CHGH0 - A CHANGE TO TEST 67 (PARITY ERROR TEST) WAS MADE TO MAKE
THE TEST DO THE INTENDED FUNCTION. IT HAD BEEN SKIPPING
THE TEST DUE TO INCORRECT SETUP PROCEDURE.

a

.TITLE CZRSBH0 RH11-RS03/04 BFT
:COPYRIGHT 1973,1980 DIGITAL EQUIPMENT CORP.,MAYNARD,MASS.
:PROGRAM BY STANLEY HARACKIEWICZ

	SWITCH	USE
	-----	-----
:	SW15= 100000	:HALT ON ERROR
:	SW14= 40000	:LOOP ON TEST
:	SW13= 20000	:INHIBIT ERROR TYPEOUTS
:	SW12= 10000	:INHIBIT OBUFSV FROM CHANGING FOR -B- PORT ONLY
:	SW11= 4000	:INHIBIT ITERATIONS
:	SW10= 2000	:0 - BELL ON PASS COMPLETE
:		:1 - BELL ON ERROR
:	SW9= 1000	:LOOP ON ERROR
:	SW8= 400	:LOOP ON TEST IN SW<7:0>
:	0	:TRAP CATCHER FROM 0 - 776
.=	46	:HOOKS FOR ACT 11
SENDAD	52	
BIT14		

520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562 100000
563 040000
564 020000
565 010000
566 004000
567 002000
568
569 001000
570 000400
571 000000
572 000046
573 000046 022172
574 000052
575 000052 040000

Address	Op1	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	Op418	Op41
---------	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	------

611				
612	000001	N=	1	:INITALIZE FOR NEWTST
613	104000	HLT=	EMT	:SET HLT TO EMT FOR ERROR TYPEOUTS
614	177776	PS=	177776	:PROCESSOR STATUS
615	177776	PSW=	PS	:PROCESSOR STATUS WORD
616	000007	BELL=	7	:BELL
617	000000	R0=	%0	:R0 - DEFINE REGISTERS
618	000001	R1=	%1	:R1
619	000002	R2=	%2	:R2
620	000003	R3=	%3	:R3
621	000004	R4=	%4	:R4
622	000005	R5=	%5	:R5
623	000006	SP=	%6	:R6 - STACK POINTER
624	000007	PC=	%7	:R7 - PROGRAM COUNTER
625	000001	BIT0=	1	:BIT EQUATES
626	000002	BIT1=	2	
627	000004	BIT2=	4	
628	000010	BIT3=	10	
629	000020	BIT4=	20	
630	000040	BIT5=	40	
631	000100	BIT6=	100	
632	000200	BIT7=	200	
633	000400	BIT8=	400	
634	001000	BIT9=	1000	
635	002000	BIT10=	2000	
636	004000	BIT11=	4000	
637	010000	BIT12=	10000	
638	020000	BIT13=	20000	
639	040000	BIT14=	40000	
640	100000	BIT15=	100000	
641	000001	GOOD=	%1	:FOR GOOD DATA
642	000000	BAD=	%0	:FOR BAD DATA
643				


```
644      001000      . =      1000
645
646 001000 000000      ICNT: 0      ;LH = ITERATION COUNT ;RH = TEST NO.
647 001002 000000      ERRORS: 0      ;ERROR COUNT
648 001004 000000 000000      PCNT: 0,0      ;2 WORD PASS COUNT
649 001010 000000      LAD: 0      ;LOOP ADDRESS FOR SCOPE
650 001012 000000      HLTADR: 0      ;ADDRESS OF LAST HLT INSTRUCTION EXECUTED
651 001014 001000      FILCHR: 1000      ;FILCHR=0 (CHAR) ;FILCHR+1=2 (COUNT)
652 001016 177564      TPS: 177564      ;OUTPUT STATUS REGISTER
653 001020 177560      TKS: 177560
654 001022 177562      TKB: 177562
655 001024 177566      TPB: 177566
656 001026 177570      SWR: 177570
657 001030 177570      DISPLAY: 177570
658
659      001100      . =      1100
660 001100 000000      SAVBAD: 0      ;LOC FOR ILLEGAL FUNCTION CODE
661 001102 000000      OBUFSV: 0      ;LOC OF OUTBUF
662
663      ;DISK I/O REGISTERS
664
665 001104 172040      RSCS1: 172040      ;DISK CONTROL + STATUS REGISTER
666 001106 172050      RSCS2: 172050      ;DISK CONTROL + STATUS REGISTER
667 001110 172042      RSWC: 172042      ;WORD COUNT REGISTER
668 001112 172044      RSBA: 172044      ;BUS ADDRESS
669 001114 172046      RSDA: 172046      ;DISK ADDRESS (DESIRED ADDRESS)
670 001116 172052      RSDS: 172052      ;DRIVE STATUS
671 001120 172054      RSER: 172054      ;ERROR REG.
672 001122 172056      RSAS: 172056      ;ATTENTION SUMMARY
673 001124 172060      RSLA: 172060      ;LOOK AHEAD
674 001126 172062      RSDB: 172062      ;DATA BUFFER REGISTER
675 001130 172064      RSMR: 172064      ;MAINTENANCE REGISTER
676 001132 172066      RSDT: 172066      ;DRIVE TYPE REGISTER
677 001134 000204      RSVEC: 204      ;INTERUPT VECTOR
678 001136 000206      RSVCPS: 206      ;INTERUPT PRIO. VECTOR
679 001140 172041      RSCS1B: 172041      ;ODD BYTE ADD FOR CS1
680 001142 172051      RSCS2B: 172051      ;ODD BYTE ADD FOR CS2
681 001144 172043      RSWCB: 172043      ;ODD BYTE ADD FOR CW
682 001146 172045      RSBAB: 172045      ;ODD BYTE ADD FOR BA
683
684      ;MEMORY MANAGEMENT REGISTER ASSIGNMENTS
685
686      177572      SR0=177572
687      172340      KIPAR0=172340
688      172342      KIPAR1=172342
689      172344      KIPAR2=172344
690      172356      KIPAR7=172356
691      172300      KIPDR0=172300
692      172302      KIPDR1=172302
693      172304      KIPDR2=172304
694      172316      KIPDR7=172316
695      000006      RW=6
696      000000      UP=00
697
```


698 :BIT ASSIGNMENTS FOR ERROR TYPEOUTS
699 :THE RS REGISTERS ARE DIVIDED INTO 3 GROUPS.
700 :CS1,CS2 AND ER ARE IN THE FIRST GROUP.THIS GROUP IS ALWAYS
701 :TYPED WITH EITHER OF THE OTHER GROUPS. AS,BA,DA, WC AND DS
702 :ARE IN THE SECOND GROUP. DT,DB,MR, AND LA ARE IN THE 3RD
703 :GROUP.YOU CAN NOT INTERMIX GROUP 2 OR 3. THEY HAVE
704 :TO BE TYPED SEPERATELY.
705 :EXAMPLE: HLT !CS1,AS,BA
706 : HLT !CS1!DT!DB
707 :

708	000001	CS1=1	:CONTROL AND STATUS 1
709	000002	ER=2	:CONTROL AND STATUS 2
710	000004	DA=4	:DESIRED ADD
711	000010	WC=10	:WORD COUNT
712	000020	BA=20	:BUS ADDRESS
713	000040	DS=40	:DRIVE STATUS
714	000100	AS=100	:ATTENTION SUMMARY
715	000200	CS2=200	:CONTROL AND STATUS REG
716	000204	LA=204	:LOOK AHEAD
717	000210	DB=210	:DATA BUFFER
718	000220	MR=220	:MAINTENANCE
719	000240	DT=240	:DRIVE TYPE

:BIT ASSIGNMENTS FOR THE REGISTER BITS

720			
721			
722			
723	040000	TRE=40000	:TRANSFER ERROR CS1
724	100000	SC=100000	:SPECIAL CONDITIONS CS1
725	000100	IR=100	:INPUT READY CS2
726	000200	OR=200	:OUTPUT READY CS2
727	002000	PGE=2000	:PROGRAM ERROR-CS2
728	010000	NED=10000	:NON-EXISTENT DRIVE CS2
729	040000	WCE=40000	:WRITE CHECK ERROR-CS2
730	100000	DLT=100000	:DATA LATE ERROR CS2
731	000200	DRY=200	:DRIVE READY DS
732	020000	PIP=20000	:POSITIONING IN PROGRESS DS
733	002000	LBT=2000	:LAST BLOCK TRANSFER-DS
734	040000	ERR=40000	:ERROR DS
735	100000	ATA=100000	:ATTENTION ACTIVE-DS
736	001000	DAO=1000	:DISK OVERFLOW ERROR-ER
737	100000	DCK=100000	:DATA CHECK ERROR-ER
738	000010	BAI=10	:BUS ADDR INCREMENT INHIBIT
739	000100	IE=100	:INTERRUPT INABLE CS1


```
740 ;WORKING LOCATIONS
741
742 001150 146723 RANNU: 146723 ;RANDOM NUMBER PRIME
743 001152 000000 UNNUM: 0 ;UNIT CURRENTLY BEING TESTED
744 001154 000000 UNITSV: 0 ;SET BIT=UNIT ON BUS
745 001156 000000 UNCMP: 0 ;FOR COMPARING FOR # OF DEVICE
746 001160 000000 ONCEE: 0 ;DID WE TEST ANY DRIVES
747 001162 000000 RS04DT: 0 ;CLR IF RS03 SET IF RS04
748 001164 000000 TIMSV: 0 ;SAVE LOC FOR TIME
749 001166 000000 AOB1: 0 ;PORT SWITCH
750 000004 WWP=4 ;WRITE WRONG PARITY
751 172100 MPRO=172100 ;PARITY REG
752 001170 000000 BPORTT: 0 ;BUFFER ADDR FOR -B- PORT
753 001172 000000 SAVEE: 0 ;WORK LOC
754 001174 177746 CACHE: 177746 ;CACHE REG
755
756 ;DISCRIPTION OF ONCEE BITS
757 ;BIT0 MEANS FOUND DRIVE
758 ;BIT11 DO TKSEL TEST
759 ;BIT12 TYPE COULD NOT FIND NED ONLY ONCE
760 ;BIT13 TYPE NO MEM ON B PORT ONLY ONCE
761 ;BIT14 0- DO WCE WITH 0 -1 DO WCE WITH 1
762 ;BIT15 MEANS ERROR FOUND
763
764 ;RH11 WORK REGISTERS
765 ;(CAN BE CHANGED IN ANY ROUTINE)
766 001176 000000 WORK: 0
767 001200 000000 WORK1: 0
768 001202 000000 WORK2: 0
769
770 001204 012706 000500 BEGIN: MOV #500,SP ;SET STACK TO *** 500 ***
771 001210 012737 023134 000024 MOV #.POWER,a#24 ;SET UP PF VECTOR
772 001216 012737 000340 000026 MOV #340,a#26 ;LOCK OUT THE WORLD
773 001224 012737 022562 000030 MOV #.HLT,a#30 ;SET EMT VECTOR
774 001232 012737 000340 000032 MOV #340,a#32 ;LOCK UP
775 001240 012737 023566 000034 MOV #.TRAP,a#34 ;SET TRAP VECTOR
776 001246 012737 000340 000036 MOV #340,a#36 ;LOCK UP
777 001254 005037 001000 CLR ICNT ;INIT ICNT
778 001260 005037 001010 CLR LAD ;INIT LAD
779 001264 042737 143777 001160 BIC #143777,ONCEE ;CLEAR ONCEE
780 001272 012737 027530 001102 MOV #OUTBUF,OBUSV ;SAVE LOC OF OUTBUFFER
781 001300 032737 004000 001160 BIT #BIT11,ONCEE ;DO TKSEL TEST?
782 001306 001402 BEQ .+6 ;NO
783 001310 000137 024160 JMP a#TKSEL ;YES
784 001314 104420 SUSWR ;SIZE FOR HDWR SWR
785 001316 005227 177777 INC #-1 ;SO TITLE PRINTOUT 1ST PASS ONLY
786 001322 001025 BNE AROUND ;THRU FOR PASSES >1
787 ;*****
788 ; ROUTINE TO PRINT DIAGNOSTICS TITLE
789 ;*****
790 001324 104402 001330 TYPE .,+2 ;.ASCIZ 'CZRSBH0 RH11-RS03/04 BFT DIAGNOSTIC'<15><12>
791 001376 000240 AROUND: NOP ;FALL THUR
792
```



```
793 ;NOW TEST FOR DRIVES
794 ;*****
795 ;TEST 1 TEST FOR DRIVES
796 ;*****
797 TST1: SCOPE
798
799 001400 104400
800 001402 012701 000010 MULTII: MOV #8,R1 ;PUT 8 INTO R1 FOR COUNT
801 001406 005077 177474 CLR @RSCS2 ;SET DEVICE TO ZERO
802 001412 012777 000007 177500 TRY: MOV #7,@RSER ;CAUSE AN ERROR +SETS BIT IN RSAS REG
803 001420 005301 DEC R1 ;DO A MAXIMUM OF 8 TIMES
804 001422 001403 BEQ DVNUM ;TESTED FOR ALL DRIVES GET OUT
805 001424 005277 177456 INC @RSCS2 ;INCREMENT DRIVE UNIT
806 001430 000770 BR TRY ;REPEAT FOR NEXT DRIVE
807 001432 017737 177464 001154 DVNUM: MOV @RSAS,UNITSV ;SAVE
808 001440 012737 000401 001156 MOV #401,UNCMP ;SETUP TO CMP WITH UNITSV
809 001446 012737 000000 001152 MOV #0,UNNUM ;PUT 0 INTO UNIT NO.
810 001454 032777 020000 177344 BIT #BIT13,@SWR ;INHIBIT TYPE OUT?
811 001462 001015 BNE STTEST ;YES
812 001464 104402 001470 TYPE ;ASCIZ <15><12>'TESTING UNIT '
813 001510 042737 100000 001160 BIC #BIT15,ONCEE ;CLEAR ERROR FLAG
814 001516 033737 001156 001154 STTEST: BIT UNCMP,UNITSV ;IS THIS DRIVE ON THE SYSTEM
815 001524 001475 BEQ TRYNX ;NO
816 001526 013777 001152 177352 MOV UNNUM,@RSCS2 ;YES PUT UNIT # INTO CS2
817 001534 022777 000004 177370 CMP #4,@RSDT ;IS THIS A RS03LA?
818 001542 001004 BNE 7$ ;NO
819 001544 012737 000004 001162 MOV #4,RS04DT ;SETUP DRIVE TYPE FOR A LA DISK
820 001552 000435 BR 1$ ;GET OUT
821 001554 022777 000000 177350 7$: CMP #0,@RSDT ;IS THIS A RS03?
822 001562 001003 BNE 2$ ;NO
823 001564 005037 001162 CLR RS04DT ;YES
824 001570 000426 BR 1$ ;GET OUT
825 001572 022777 000001 177332 2$: CMP #1,@RSDT ;IS THIS A RS03 4US?
826 001600 001003 BNE 3$ ;NO
827 001602 005037 001162 CLR RS04DT ;RS03
828 001606 000417 BR 1$ ;GET OUT
829 001610 022777 000002 177314 3$: CMP #2,@RSDT ;IS THIS A RS04?
830 001616 001410 BEQ 6$ ;YES
831 001620 022777 000003 177304 CMP #3,@RSDT ;IS IT A RS04?
832 001626 001404 BEQ 6$ ;YES
833 001630 043737 001156 001154 BIC UNCMP,UNITSV ;CLEAR UNWANTED ATA BIT
834 001636 000430 BR TRYNX ;GET A NEW NUMBER
835 001640 052737 177777 001162 6$: BIS #-1,RS04DT ;YES RS04
836 001646 032777 020000 177152 1$: BIT #BIT13,@SWR ;INHIBIT TYPE OUT?
837 001654 001020 BNE 4$ ;YES
838 001656 032737 100000 001160 BIT #BIT15,ONCEE ;ANY ERRORS?
839 001664 001404 BEQ 5$ ;NO
840 001666 104402 001672 TYPE ;ASCIZ <15><12><12>
841 ;*****4
842 5$: MOV UNNUM,-(6) ;PUT UNNUM ON STACK
843 001676 013746 001152 TYPES ;TYPE STACK IN OCTAL - SUPRESS
844 001702 104406 ;*****
845 001704 104402 000040 TYPE ,40 ;TYPE SPACE
846 001710 042737 100000 001160 BIC #BIT15,ONCEE ;CLEAR ERROR FLAG
847 001716 000473 4$: BR NOWGO ;NOW TEST
```



```
848 001720 006337 001156      TRYNX: ASL      UNCMP      ;CHECK NEXT BIT FOR DRIVE
849 001724 103403              BCS      CHCKDV      ;DID WE TEST ANY REG?
850 001726 005237 001152      INC      UNNUM      ;INC UNIT #
851 001732 000671              BR       STTEST      ;CHECK FOR NEXT DRIVE
852
853 001734 032737 000001 001160 CHCKDV: BIT      #BIT0,ONCEE ;DID WE TEST ANY DRIVES?
854 001742 001057              BNE      DONEE      ;YES WE DID TEST A DRIVE
855 001744 012737 100000 001156 MOV      #100000,UNCMP ;NO DRIVES TESTED, COULD NOT SET
856 001752 005037 001152      CLR      UNNUM      ;ANY AS BITS, THUS DEFAULTS TO
857 001756 032777 020000 177042 BIT      #BIT13,ASWR ;INHIBIT TYPE OUT?
858 001764 001045              BNE      4$         ;YES
859 001766 013746 001152      MOV      UNNUM,-(6) ;PUT UNNUM ON STACK
860 001772 104406              TYPES             ;TYPE STACK IN OCTAL - SUPRESS
861 001774 104402 000040      TYPE      ,40      ;TYPE SPACE
862 002000 104402 002004      TYPE      ,.+2     ;ASCIZ <15><12>'COULD NOT FIND DRIVE WILL TEST DRIVE 0
863 002076 000000              HALT             ;WAIT
864 002100 000402              4$: BR       NOWGO    ;TEST DRIVE 0
865 002102 000137 022126      DONEE: JMP      DONE     ;GET OUT
866
867      ;THIS TEST IS DESIGNED TO TEST THE ABILITY OF RESET
868      ;TO CLEAR ALL THE RH AND RS REGISTERS
869
870 002106 012737 027530 001102 NOWGO: MOV      #OUTBUF,OBUSV ;SAVE LOC OF OUTBUFFER
871 002114 052737 000001 001160 BIS      #BIT0,ONCEE ;SET FOUND DRIVE FLAG
872 002122 013737 022560 001164 MOV      TIMES,TIMSV ;SAVE TIME
873 002130 012737 000001 022560 MOV      #1,TIMES ;ONLY TEST ONCE
874
875      ;*****
876      ;TEST 2      RESET TEST FOR REGISTERS
877      ;*****
877 002136 104400      TST2: SCOPE
878 002140 012737 000340 177776 MOV      #340,@#PS ;LOCK OUT INTERRUPTS
879 002146 013777 001152 176732 MOV      UNNUM,@RSCS2 ;GET UNIT #
880 002154 012777 177776 176722 MOV      #177776,@RSCS1 ;SET ALL
881 002162 012777 177777 176722 MOV      #177777,@RSBA ;POSSIBLE R/W
882 002170 012777 177777 176716 MOV      #177777,@RSDA ;BITS IN THESE REGISTERS
883 002176 012777 177777 176714 MOV      #177777,@RSER
884 002204 012777 177777 176716 MOV      #177777,@RSMR
885 002212 012777 177777 176670 MOV      #177777,@RSWC
886 002220 012777 177737 176660 MOV      #177737,@RSCS2
887 002226 000005      RESET ;CLEAR ALL BITS IN ALL REG.
888
889      ;TEST RSCS2 FOR CLEARED BITS
890
891 002230 022777 000100 176650 CMP      #100,@RSCS2 ;DID THESE BITS GET CLEARED?
892 002236 001401      BEQ      ,+4 ;YES
893 002240 104200      HLT      !CS2 ;(417) SHOULD BE CLEARED IN CS2
894 002242 013777 001152 176636 MOV      UNNUM,@RSCS2 ;PUT # OF UNIT IN TEST IN CS2
895 002250 022777 010600 176640 CMP      #10600,@RSDS ;IS DPR AND MOL SET?
896 002256 001401      BEQ      ,+4 ;YES
897 002260 104040      HLT      !DS ;NO WHY NOT?
898
899 002262 022777 004200 176614 ;TEST CONTROL AND STATUS REG 1
900 002270 001401      CMP      #4200,@RSCS1 ;DID THE READY BIT SET?
901 002272 104001      BEQ      ,+4 ;YES
          HLT      !CS1 ;READY SHOULD BE SET
```



```

902      ;TEST BUS ADDRESS REGISTER
903
904 002274 005777 176612      TST      @RSBA      ;IS BA REG. CLEARED
905 002300 001401      BEQ      .+4      ;YES
906 002302 104020      HLT      !BA      ;SHOULD BE 0
907
908      ;TEST DISK ADDRESS REGISTER
909
910 002304 005777 176604      TST      @RSDA      ;IS DA CLEARED
911 002310 001401      BEQ      .+4      ;YES
912 002312 104004      HLT      !DA      ;SHOULD BE 0
913
914      ;TEST ERROR REG RSER
915
916 002314 005777 176600      TST      @RSER      ;DID RSER CLEAR?
917 002320 001401      BEQ      .+4      ;YES
918 002322 104002      HLT      !ER      ;BITS(157015) SHOULD BE CLEARED
919
920      ;TEST RS MAINTENANCE REGISTER
921
922 002324 032777 000077 176576      BIT      #77,@RSMR      ;DID THESE BITS GET CLEARED
923 002332 001401      BEQ      .+4      ;YES
924 002334 104220      HLT      !MR      ;BITS(77) SHOULD BE 0
925
926      ;TEST WC REG IT SHOULD NOT CHANGE
927
928 002336 022777 177777 176544      CMP      #177777,@RSWC      ;DID IT CHANGE?
929 002344 001401      BEQ      .+4      ;NO
930 002346 104010      HLT      !WC      ;RESET SHOULD NOT MODIFY RSWC
931
932      ;TEST RSAS
933
934 002350 005777 176546      TST      @RSAS      ;IS REG CLEAR
935 002354 001401      BEQ      .+4      ;YES
936 002356 104100      HLT      !AS      ;NO

```



```

937 ;*****
938 ;TEST 3 TEST CLEAR BIT IN CS2 ON ALL THE R/W BITS
939 ;*****
940 002360 104400 TST3: SCOPE
941
942 002362 012737 000340 177776 TTAGG: MOV #340,@#PS ;LOCK OUT INTERRUPTS
943 002370 013777 001152 176510 MOV UNNUM,@RSCS2 ;GET UNIT #
944 002376 012777 043576 176500 MOV #43576,@RSCS1 ;SET ALL
945 002404 012777 177777 176500 MOV #177777,@RSBA ;POSSIBLE
946 002412 012777 177777 176474 MOV #177777,@RSDA ;REGISTERS
947 002420 012777 177777 176472 MOV #177777,@RSER
948 002426 012777 177777 176472 MOV #177777,@RSDB
949 002434 012777 177777 176446 MOV #177777,@RSWC
950 002442 012777 020417 176436 MOV #20417,@RSCS2
951 002450 012777 000071 176452 MOV #71,@RSMR
952 002456 012777 000040 176422 MOV #40,@RSCS2 ;CLEAR ALL BITS
953 002464 022777 000100 176414 CMP #100,@RSCS2 ;DID THE RIGHT BITS CLEAR?
954 002472 001401 BEQ .+4 ;YES
955 002474 104200 HLT !CS2 ; (417) SHOULD BE CLEARED IN CS2
956 002476 013777 001152 176402 MOV UNNUM,@RSCS2 ;GET DRIVE NUMBER
957 002504 032777 173577 176372 BIT #173577,@RSCS1 ;DID ALL BITS GET CLEARED
958 002512 001401 BEQ .+4 ;YES
959 002514 104001 HLT !CS1 ;NO, ALL BITS SHOULD BE 0
960 ;TEST BUS ADDRESS REGISTER
961
962 002516 005777 176370 TST @RSBA ;IS BA REG. CLEARED
963 002522 001401 BEQ .+4 ;YES
964 002524 104020 HLT !BA ;SHOULD BE 0
965
966 ;TEST DISK ADDRESS REGISTER
967
968 002526 005777 176362 TST @RSDA ;IS DA CLEARED
969 002532 001401 BEQ .+4 ;YES
970 002534 104020 HLT !BA ;SHOULD BE 0
971
972 ;TEST ERROR REG RSER
973
974 002536 005777 176356 TST @RSER ;DID THESE BITS GET CLEARED
975 002542 001401 BEQ .+4 ;YES
976 002544 104002 HLT !ER ;BITS(157015) SHOULD BE CLEARED
977
978 ;TEST RS MAINTENANCE REGISTER
979 002546 032777 000077 176354 BIT #77,@RSMR ;DID THESE BITS GET CLEARED
980 002554 001401 BEQ .+4 ;YES
981 002556 104220 HLT !MR ;BITS(77) SHOULD BE 0
982
983 ;TEST WC REG. IT SHOULD NOT CHANGE
984 002560 022777 177777 176322 CMP #177777,@RSWC ;DID WC CHANGE
985 002566 001401 BEQ .+4 ;NO
986 002570 104010 HLT !WC ;WHY DID IT CHANGE?

```



```

987
988
989
990 002572 104400
991
992
993
994 002574 104414
995 002576 013737 001164 022560
996 002604 012777 003576 176272
997 002612 022777 007776 176264
998 002620 001401
999 002622 104001
1000 002624 012777 002524 176252
1001 002632 022777 006724 176244
1002 002640 001401
1003 002642 104001
1004 002644 012777 001052 176232
1005 002652 022777 005252 176224
1006 002660 001401
1007 002662 104001
1008 002664 104400
1009
1010
1011 002666 012777 043576 176210
1012 002674 005077 176204
1013 002700 022777 004200 176176
1014 002706 001401
1015 002710 104001
1016
1017
1018
1019
1020 002712 104400
1021
1022 002714 000005
1023 002716 022777 000100 176162
1024 002724 001401
1025 002726 104200
1026 002730 012777 021037 176150
1027 002736 022777 021137 176142
1028 002744 001405
1029 002746 017700 176134
1030 002752 012701 021137
1031 002756 104000

;*****
;TEST 4 SET AND CLEAR ALL REGISTERS
;*****
TST4: SCOPE
;CAN WE SET THE FUNCTION BITS IN THE RSCS1 REG.
;BITS 7,6,5,4,3,2&1

CLRDK
MOV TIMSV,TIMES ;CLEAR ALL RS REG
MOV #3576,@RSCS1 ;GET TIME
CMP #7776,@RSCS1 ;SET DISK FUNCTION BITS
BEQ +4 ;ARE THESE BITS SET?
HLT !CS1 ;NO
MOV #2524,@RSCS1 ;SHOULD = 7776
CMP #6724,@RSCS1 ;SET THESE BITS
BEQ +4 ;DID THEY SET
HLT !CS1 ;YES
MOV #1052,@RSCS1 ;SHOULD BE 6724
CMP #5252,@RSCS1 ;SET THESE BITS
BEQ +4 ;ARE THEY =?
HLT !CS1 ;YES
;SHOULD = 5252

TST5: SCOPE
;CLEAR THE FUNCTION BITS

MOV #43576,@RSCS1 ;SET DISK FUNCTION BITS
CLR @RSCS1
CMP #4200,@RSCS1 ;IS THE READY BIT SET
BEQ +4 ;YES
HLT !CS1 ;RSCS1 SHOULD = 4200

;*****
;TEST 6 TEST RSCS2
;*****
TST6: SCOPE

RESET
CMP #100,@RSCS2 ;CLEAR WORLD
BEQ +4 ;DID THEY CLEAR?
HLT !CS2 ;YES
MOV #21037,@RSCS2 ;NO
CMP #21137,@RSCS2 ;SET BITS 21017
BEQ 1$ ;DID THESE BITS GET SET
MOV @RSCS2,BAD ;YES
MOV #21137,GOOD ;WHAT CS2 SHOULD =
HLT ;CS2 = BAD GOOD = CORRECT ANS

```



```
1032 002760 012777 020025 176120 1$: MOV #20025,@RSCS2 ;SET THESE BITS
1033 002766 022777 020125 176112 CMP #20125,@RSCS2 ;DID THESE BITS GET SET
1034 002774 001401 BEQ .+4 ;YES
1035 002776 104200 HLT !CS2 ;NO,CS2 SHOULD = 20125
1036 003000 012777 000012 176100 MOV #12,@RSCS2 ;LOAD THESE BITS
1037 003006 022777 000112 176072 CMP #112,@RSCS2 ;DID THESE BITS GET SET IN CS2
1038 003014 001401 BEQ .+4 ;YES
1039 003016 104200 HLT !CS2 ;BAD = CS2 GOOD = CORRECT ANS
1040 003020 012777 177777 176060 MOV #-1,@RSCS2 ;SET BITS
1041 003026 005077 176054 CLR @RSCS2 ;CLEAR THEM
1042 003032 022777 000100 176046 CMP #100,@RSCS2 ;DID CLEAR WORK
1043 003040 001401 BEQ .+4 ;YES
1044 003042 104200 HLT !CS2 ;R/W BITS DID NOT CLEAR
1045 003044 013777 001152 176034 MOV UNNUM,@RSCS2 ;GET UNIT #
1046 003052 104400
1047
1048 TST7: SCOPE
;CAN WE SET ALL THE RSBA BITS
1049 003054 012777 177777 176030 MOV #177777,@RSBA ;SET THE BITS
1050 003062 022777 177776 176022 CMP #177776,@RSBA ;DID THEY SET
1051 003070 001401 BEQ .+4 ;YES
1052 003072 104020 HLT !BA ;BITS 17776 SHOULD BE SET
1053 003074 012777 125252 176010 MOV #125252,@RSBA ;SET THESE BITS
1054 003102 022777 125252 176002 CMP #125252,@RSBA ;ARE THEY =
1055 003110 001401 BEQ .+4 ;YES
1056 003112 104020 HLT !BA ;SHOULD BE 125252
1057 003114 012777 052524 175770 MOV #52524,@RSBA ;SET THESE BITS
1058 003122 022777 052524 175762 CMP #52524,@RSBA ;ARE THEY =
1059 003130 001401 BEQ .+4 ;YES
1060 003132 104020 HLT !BA ;SHOULD BE 52524
1061
1062 003134 104400 TST10: SCOPE
1063 ;FLOAT A 1 THROUGH RSBA
1064
1065 003136 012701 000002 FLOTBA: MOV #2,GOOD ;GET A 2
1066 003142 000241 CLC ;CLEAR CARRY
1067 003144 010177 175742 1$: MOV GOOD,@RSBA ;FLOAT NUMBER
1068 003150 017700 175736 MOV @RSBA,BAD ;GET BA
1069 003154 020100 CMP GOOD,BAD ;COMPARE BA
1070 003156 001401 BEQ .+4 ;BA CORRECT
1071 003160 104000 HLT ;BAD=BA GOOD=CORRECT ANS
1072 003162 006101 ROL GOOD ;ROTATE NUMBER
1073 003164 103367 BCC 1$ ;LOOP TILL DONE
```



```
1074 003166 104400      TST11:  SCOPE
1075
1076                      ;CLEAR THE RSBA REGISTER
1077
1078 003170 012777 177777 175714      MOV    #177777,@RSBA    ;SET RSBA EQUAL TO ALL ONES
1079 003176 005077 175710      CLR    @RSBA
1080 003202 005777 175704      TST    @RSBA    ;TEST FOR BIT0 SET IN RSBA (READ ONLY BIT)
1081 003206 001401      BEQ    .+4    ;YES
1082 003210 104020      HLT    !BA    ;NO
1083 003212 104400      TST12:  SCOPE
1084
1085                      ;CAN WE SET ALL BITS IN RSWC REGISTER
1086
1087 003214 012777 177777 175666      MOV    #177777,@RSWC    ;SET WC BITS
1088 003222 022777 177777 175660      CMP    #177777,@RSWC    ;ARE ALL BITS SET
1089 003230 001401      BEQ    .+4    ;YES
1090 003232 104010      HLT    !WC    ;NO
1091 003234 012777 125252 175646      MOV    #125252,@RSWC    ;SET THESE BITS
1092 003242 022777 125252 175640      CMP    #125252,@RSWC    ;ARE THEY =
1093 003250 001401      BEQ    .+4    ;YES
1094 003252 104010      HLT    !WC    ;SHOULD BE 125252
1095 003254 012777 052525 175626      MOV    #52525,@RSWC    ;SET THESE BITS
1096 003262 022777 052525 175620      CMP    #52525,@RSWC    ;ARE THEY =
1097 003270 001401      BEQ    .+4    ;YES
1098 003272 104010      HLT    !WC    ;SHOULD BE 152525
1099 003274 104400      TST13:  SCOPE
1100
1101                      ;FLOAT A 1 THROUGH RSWC
1102
1103 003276 012701 000001      FLOTWC: MOV    #1,GOOD    ;GET A 1
1104 003302 000241      CLC                      ;CLEAR CARRY
1105 003304 010177 175600      1$:  MOV    GOOD,@RSWC    ;FLOAT NUMBER
1106 003310 017700 175574      MOV    @RSWC,BAD    ;GET WC
1107 003314 020100      CMP    GOOD,BAD    ;COMPARE WC
1108 003316 001401      BEQ    .+4    ;WC CORRECT
1109 003320 104000      HLT                      ;BAD=WC GOOD=CORRECT ANS
1110 003322 006101      ROL    GOOD    ;ROTATE NUMBER
1111 003324 103367      BCC    1$    ;LOOP TILL DONE
```



```
1112 ;CLEAR THE WORD COUNT REGISTER
1113 003326 104400 TST14: SCOPE
1114
1115 003330 012777 177777 175552 MOV #177777,@RSWC ;SET RSWC REGISTER EQUAL TO ALL ONES
1116 003336 005077 175546 CLR @RSWC
1117 003342 005777 175542 TST @RSWC ;DID ALL BITS GET CLEARED
1118 003346 001401 BEQ .+4 ;YES
1119 003350 104010 HLT !WC ;NO
1120 003352 104400 TST15: SCOPE
1121
1122 ;CAN WE SET ALL THE BITS IN THE RSDA REGISTER.
1123
1124 003354 012777 177777 175532 MOV #177777,@RSDA ;SET ALL BITS
1125 003362 022777 177777 175524 CMP #177777,@RSDA ;ARE THE BITS SET
1126 003370 001401 BEQ .+4 ;YES
1127 003372 104004 HLT !DA ;NO
1128 003374 012777 125252 175512 MOV #125252,@RSDA ;SET THESE BITS
1129 003402 022777 125252 175504 CMP #125252,@RSDA ;ARE THEY =
1130 003410 001401 BEQ .+4 ;YES
1131 003412 104004 HLT !DA ;SHOULD BE 125252
1132 003414 012777 052525 175472 MOV #52525,@RSDA ;SET THESE BITS
1133 003422 022777 052525 175464 CMP #52525,@RSDA ;ARE THEY =
1134 003430 001401 BEQ .+4 ;YES
1135 003432 104004 HLT !DA ;SHOULD BE 52525
1136 003434 104400 TST16: SCOPE
1137
1138 ;FLOAT A 1 THROUGH RSDA
1139
1140 003436 012701 000001 FLOTDA: MOV #1,GOOD ;GET A 1
1141 003442 000241 CLC ;CLEAR CARRY
1142 003444 010177 175444 1$ : MOV GOOD,@RSDA ;FLOAT NUMBER
1143 003450 017700 175440 MOV @RSDA,BAD ;GET DA
1144 003454 020100 CMP GOOD,BAD ;COMPARE DA
1145 003456 001401 BEQ .+4 ;DA CORRECT
1146 003460 104000 HLT ;BAD=DA GOOD=CORRECT ANS
1147 003462 006101 ROL GOOD ;ROTATE NUMBER
1148 003464 103367 BCC 1$ ;LOOP TILL DONE
```



```

1149
1150 003466 104400
1151
1152 003470 012777 177777 175416
1153 003476 005077 175412
1154 003502 005777 175406
1155 003506 001401
1156 003510 104004
1157 003512 104400
1158
1159
1160
1161 003514 012777 177777 175376
1162 003522 022777 177017 175370
1163 003530 001401
1164 003532 104002
1165 003534 112777 000001 175356
1166 003542 022777 000001 175350
1167 003550 001401
1168 003552 104002
1169
1170 003554 104400
1171
1172 003556 012777 052005 175334
1173 003564 022777 052005 175326
1174 003572 001401
1175 003574 104002
1176 003576 104400
1177
1178 003600 012777 125012 175312
1179 003606 022777 125012 175304
1180 003614 001401
1181 003616 104002

;CAN WE CLEAR THE RSDA REG.
TST17: SCOPE

MOV #177777,@RSDA ;SET RSDA TO ALL ONES
CLR @RSDA
TST @RSDA ;TEST FOR ZERO RSDA
BEQ +4 ;YES
HLT !DA ;ANS SHOULD BE 0

TST20: SCOPE

;SET AND CLEAR THE RSER REG.

MOV #177777,@RSER ;SET THESE BITS
CMP #177017,@RSER ;DID THEY SET
BEQ +4 ;YES
HLT !ER ;RSER SHOULD = 157017
MOVB #1,@RSER ;A MOVB INST
CMP #1,@RSER ;SHOULD MODIFY COMPLETE WD
BEQ +4 ;OK
HLT !ER

TST21: SCOPE

MOV #52005,@RSER ;SET THESE BITS
CMP #52005,@RSER ;DID THEY SET
BEQ +4 ;YES
HLT !ER ;ER SHOULD = 52005

TST22: SCOPE

MOV #125012,@RSER ;SET THESE BITS
CMP #125012,@RSER ;DID THEY SET
BEQ +4 ;YES
HLT !ER ;ER SHOULD = 105012

```



```
1182 003620 104400          TST23: SCOPE
1183
1184 003622 012777 177017 175270      MOV    #177017,@RSER      ;SET THESE BITS
1185 003630 005077 175264              CLR    @RSER          ;CLEAR THEM
1186 003634 005777 175260              TST    @RSER          ;DID THEY CLEAR
1187 003640 001401              BEQ     .+4              ;YES
1188 003642 104002              HLT     !ER              ;SHOULD = 0
1189 003644 104400          TST24: SCOPE
1190
1191          ;SET AND CLEAR RSMR
1192
1193 003646 012777 000070 175254      MOV    #70,@RSMR      ;SET THESE BITS
1194 003654 017737 175250 001176      MOV    @RSMR,WORK    ;PUT INTO WORKABLE REG
1195 003662 042737 177700 001176      BIC    #177700,WORK  ;CLEAR JUNK
1196 003670 022737 000070 001176      CMP    #70,WORK     ;DID THEY SET
1197 003676 001401              BEQ     .+4              ;YES
1198 003700 104220              HLT     !MR              ;SHOULD = 70
1199 003702 104400          TST25: SCOPE
1200
1201 003704 012777 000070 175216      MOV    #70,@RSMR      ;SET BITS
1202 003712 005077 175212              CLR    @RSMR          ;CLEAR THEM
1203 003716 032777 000077 175204      BIT    #77,@RSMR    ;DID THEY CLEAR
1204 003724 001401              BEQ     .+4              ;YES
1205 003726 104220              HLT     !MR              ;BITS (77) SHOULD = 0
1206 003730 104400          TST26: SCOPE
1207
1208 003732 012777 000050 175170      MOV    #50,@RSMR      ;SET BITS
1209 003740 017737 175164 001176      MOV    @RSMR,WORK    ;PUT IN WORKABLE REG
1210 003746 042737 177700 001176      BIC    #177700,WORK  ;CLEAR JUNK
1211 003754 022737 000050 001176      CMP    #50,WORK     ;DID THESE BITS SET
1212 003762 001401              BEQ     .+4              ;YES
1213 003764 104220              HLT     !MR              ;BITS (50) SHOULD BE SET
1214 003766 104400          TST27: SCOPE
1215
1216 003770 012777 000020 175132      MOV    #20,@RSMR      ;SET BITS
1217 003776 017737 175126 001176      MOV    @RSMR,WORK    ;PUT INTO WORKABLE REG
1218 004004 042737 177700 001176      BIC    #177700,WORK  ;CLEAR JUNK
1219 004012 022737 000020 001176      CMP    #20,WORK     ;DID THEY SET
1220 004020 001401              BEQ     .+4              ;YES
1221 004022 104220              HLT     !MR              ;MR SHOULD AT LEAST HAVE A (21)
```



```
1222 *****
1223 ;TEST 30 LOAD RANDOM NUMBERS INTO RSWC, RSDA AND RSBA
1224 *****
1225 TST30: SCOPE
1226
1227 004024 104400
1228 004026 012737 001000 001176 RANTS: MOV #1000,WORK ;MAKE TABLE 1000 WDS LONG
1229 004034 012701 027530 MOV #OUTBUF,R1 ;GET SRARTING LOC OF TABLE
1230 004040 004537 027000 JSR R5,RANDOM ;GENERATE #
1231 004044 012704 027530 MOV #OUTBUF,R4 ;SETUP FOR COMPARE
1232 004050 012737 004056 001010 MOV #LOP1,LAD ;SETUP LOOP ADDR
1233 004056 012703 001000 LOP1: MOV #1000,R3 ;LOAD TEST COUNTER
1234 004062 005303 4$: DEC R3 ;DONE WITH COMPARE?
1235 004064 001413 BEQ 1$ ;YES
1236 004066 013705 001110 MOV RSWC,R5 ;GET WC ADDRESS
1237 004072 011415 MOV (R4),(R5) ;LOAD WC
1238 004074 021524 CMP (R5),(R4)+ ;IS IT CORRECT?
1239 004076 001771 BEQ 4$ ;YES
1240 004100 017700 175004 MOV @RSWC,BAD ;GET BAD WC
1241 004104 014401 MOV -(R4),GOOD ;GET GOOD ANS
1242 004106 104000 HLT ;TYPE THEM OUT
1243 004110 005724 TST (R4)+ ;UPDATE RANDOM NUMBER
1244 004112 000763 BR 4$ ;CONT
1245 004114 012704 027530 1$: MOV #OUTBUF,R4 ;GET STARTING LOC OF TABLE
1246 004120 012737 004126 001010 MOV #LOP2,LAD ;SETUP LOOP ADDR
1247 004126 012703 001000 LOP2: MOV #1000,R3 ;SETUP TEST COUNTER
1248 004132 005303 3$: DEC R3 ;DONE YET?
1249 004134 001413 BEQ 1$ ;YES
1250 004136 013705 001114 MOV RSDA,R5 ;LOAD DA ADDRESS INTO R5
1251 004142 011415 MOV (R4),(R5) ;LOAD DA
1252 004144 021524 CMP (R5),(R4)+ ;IS IT CORRECT?
1253 004146 001771 BEQ 3$ ;YES
1254 004150 017700 174740 MOV @RSDA,BAD ;GET BAD DATA
1255 004154 014401 MOV -(R4),GOOD ;GET GOOD DATA
1256 004156 104000 HLT ;TYPE IT OUT
1257 004160 005724 TST (R4)+ ;UPDATE RANDOM NUMBER
1258 004162 000763 BR 3$ ;CONTINUE
1259 004164 012704 027530 1$: MOV #OUTBUF,R4 ;GET STARTING LOC OF TABLE
1260 004170 012737 004176 001010 MOV #LOP3,LAD ;SETUP LOOP ADDR
1261 004176 012703 001000 LOP3: MOV #1000,R3 ;SETUP TEST COUNTER
1262 004202 005303 3$: DEC R3 ;DONE YET?
1263 004204 001416 BEQ 2$ ;YES
1264 004206 013705 001112 MOV RSBA,R5 ;LOAD ADDRESS OF BA INTO R5
1265 004212 011415 MOV (R4),(R5) ;LOAD BA
1266 004214 042714 000001 BIC #BIT0,(R4) ;CLEAR BIT 0
1267 004220 021514 CMP (R5),(R4) ;IS IT CORRECT?
1268 004222 001767 BEQ 3$ ;YES
1269 004224 017700 174662 MOV @RSBA,BAD ;GET BAD DATA
1270 004230 011401 MOV (R4),GOOD ;GET GOOD DATA
1271 004232 104000 HLT ;TYPE IT OUT
1272 004234 000400 BR 1$ ;GET OUT
1273 004236 005724 1$: TST (R4)+ ;GET NEW NUMBER
1274 004240 000760 2$: BR 3$ ;CONTINUE
1275 004242 000240 2$: NOP
```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 29
TST31 TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA

SEQ 0028

```
1275 ;*****
1276 ;TEST 31 TEST ODD BYTE INSTRUCTIONS ON CS1, CS2, WC AND BA
1277 ;*****
1278 004244 104400 TST31: SCOPE
1279
1280 004246 104414 BITST: CLRDK ;CLEAR ALL RS REG
1281 004250 012777 003566 174626 MOV #3566,@RSCS1 ;LOAD CS1
1282 004256 112777 000005 174654 MOVB #5,@RSCS1B ;LOAD BIT
1283 004264 022777 006766 174612 CMP #6766,@RSCS1 ;DID IT LOAD?
1284 004272 001401 BEQ +4 ;YES
1285 004274 104001 HLT !CS1
1286 004276 112777 000032 174600 MOVB #32,@RSCS1
1287 004304 022777 006632 174572 CMP #6632,@RSCS1
1288 004312 001401 BEQ +4
1289 004314 104001 HLT !CS1 ;CS1 SHOULD = 6632
1290
1291 004316 104400 TST32: SCOPE
1292
1293 004320 013777 001152 174560 BITCS2: MOV UNNUM,@RSCS2 ;LOAD UNIT NUMBER
1294 004326 052777 177400 174552 BIS #177400,@RSCS2 ;LOAD ALL BITS
1295 004334 105077 174602 CLRB @RSCS2B ;CLR UPPER BYTE
1296 004340 013701 001152 MOV UNNUM,GOOD ;GET UNIT NO.
1297 004344 052701 000100 BIS #100,GOOD ;SET OR BIT
1298 004350 017700 174532 MOV @RSCS2,BAD ;GET CS2
1299 004354 020001 CMP BAD,GOOD ;IS CS2 CORRECT?
1300 004356 001401 BEQ +4 ;YES
1301 004360 104000 HLT ;LOAD BYTE DID NOT WORK
1302
1303 004362 104400 TST33: SCOPE
1304
1305 004364 012777 025252 174516 BITWC: MOV #25252,@RSWC ;LOAD WC
1306 004372 112777 000377 174544 MOVB #377,@RSWCB ;LOAD BIT
1307 004400 022777 177652 174502 CMP #177652,@RSWC ;DID IT LOAD?
1308 004406 001401 BEQ +4 ;YES
1309 004410 104010 HLT !WC ;NO WC SHOULD =177652
1310 004412 112777 000123 174470 MOVB #123,@RSWC
1311 004420 022777 177523 174462 CMP #177523,@RSWC
1312 004426 001401 BEQ +4
1313 004430 104010 HLT !WC ;WC SHOULD = 177523
1314
1315 004432 104400 TST34: SCOPE
1316
1317 004434 012777 025252 174450 BITBA: MOV #25252,@RSBA ;LOAD DA
1318 004442 112777 000377 174476 MOVB #377,@RSBAB ;LOAD BIT
1319 004450 022777 177652 174434 CMP #177652,@RSBA ;DID IT LOAD?
1320 004456 001401 BEQ +4 ;YES
1321 004460 104020 HLT !BA ;DA SHOULD =177652
1322 004462 112777 000125 174422 MOVB #125,@RSBA
1323 004470 022777 177524 174414 CMP #177524,@RSBA
1324 004476 001401 BEQ +4
1325 004500 104020 HLT !BA ;BA SHOULD = 177525
1326 004502 104414 CLRDK ;CLEAR ALL RS REG
```



```
1327
1328
1329
1330 004504 104400
1331
1332
1333
1334 004506 104414
1335 004510 017700 174412
1336 004514 017700 174366
1337 004520 012701 100100
1338 004524 053701 001152
1339 004530 020001
1340 004532 001401
1341 004534 104200
1342 004536 022777 144200 174340
1343 004544 001401
1344 004546 104001
1345 004550 012777 040000 174326
1346 004556 032777 140000 174320
1347 004564 001401
1348 004566 104001
1349 004570 017700 174312
1350 004574 042701 100000
1351 004600 020100
1352 004602 001401
1353 004604 104200
1354
1355
1356
1357 004606 104400
1358
1359 004610 104414
1360 004612 005077 174310
1361 004616 012777 177777 174302
1362 004624 012737 002000 001176
1363 004632 012701 000300
1364 004636 053701 001152
1365 004642 017700 174240
1366 004646 020100
1367 004650 001404
1368 004652 005337 001176
1369 004656 001371
1370 004660 104200
1371 004662 005001
1372 004664 017700 174236
1373 004670 020100
1374 004672 001401
1375 004674 104000
1376 004676 012701 177777
1377 004702 017700 174220
1378 004706 020100
1379 004710 001401
1380 004712 104000

*****
:TEST 35 TEST DATA LATE IN CS2
*****
TST35: SCOPE

;DO A READ FROM SILO: SHOULD GET DLT + TRE ERROR BECAUSE SILO IS EMPTY
SILOB: CLRDK ;CLEAR ALL RS REG
MOV @RSDB,BAD ;READ FROM EMPTY SILO
MOV @RSCS2,BAD ;GET CS2
MOV #100100,GOOD ;GET CORRECT ANS
BIS UNNUM,GOOD ;FOR CS2
CMP BAD,GOOD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT !CS2 ;SHOULD HAVE DLT ERROR
CMP #144200,@RSCS1 ;DID SC AND TRE SET?
BEQ .+4 ;YES
HLT !CS1 ;SC AND TRE SHOULD BE SET
MOV #TRE,@RSCS1 ;CLEARS ERROR BIT
BIT #140000,@RSCS1 ;DID SC + TRE CLEAR
BEQ .+4 ;YES
HLT !CS1 ;TRE AND SC SHOULD BE 0
MOV @RSCS2,BAD ;GET CS2
BIC #BIT15,GOOD ;GET CORRECT ANS
CMP GOOD,BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT !CS2 ;DLT SHOULD BE 0
*****
:TEST 36 LOAD RSDB WITH ALL ONES AND ALL ZEROS
*****
TST36: SCOPE
ZERONE: CLRDK ;CLEAR ALL RS REG
CLR @RSDB ;LOAD DB WITH ALL 0
MOV #177777,@RSDB ;LOAD DB WITH ALL ONES
MOV #2000,WORK ;TIME OUT ROUTINE
MOV #300,GOOD ;GET CORRECT FOR CS2
BIS UNNUM,GOOD
2$: MOV @RSCS2,BAD ;GET CS2
CMP GOOD,BAD ;IS IT CORRECT?
BEQ 3$ ;YES
DEC WORK ;TO WAIT FOR OR
BNE 2$ ;TO SET
HLT !CS2 ;OR SHOULD BE SET
3$: CLR GOOD ;LOAD BAD WITH DB
MOV @RSDB,BAD ;IS BAD CORRECT
CMP GOOD,BAD ;YES
BEQ .+4 ;COULD NOT FLOAT 0 THROUGH DB
HLT ;LOAD GOOD WITH ANS
MOV #-1,GOOD ;GET DATA FROM DB
MOV @RSDB,BAD ;IS DB CORRECT
CMP GOOD,BAD ;YES
BEQ .+4 ;BAD SHOULD = 177777
HLT
```


1381	004714	104400				TST37: SCOPE	
1382						;TEST FOR 66 LOCATIONS IN SILO PUT COUNT IN EVERY LOCATION	
1383							
1384	004716	104414				SILO: CLRDK	;CLEAR ALL RS REG
1385	004720	005001				CLR R1	;CLEAR COUNTER
1386	004722	005201				1\$: INC R1	;INCREMENT COUNTER
1387	004724	010177	174176			MOV R1,@RSDB	;LOAD SILO
1388	004730	022701	000102			CMP #66.,R1	;LAST LOC. YET?
1389	004734	001372				BNE 1\$	;NO LOOP AGAIN
1390	004736	012701	000200			MOV #200,GOOD	;GET CORRECT ANS FOR CS2
1391	004742	053701	001152			BIS UNNUM,GOOD	
1392	004746	017700	174134			MOV @RSCS2,BAD	;GET CS2
1393	004752	020100				CMP GOOD,BAD	;IS CS2 CORRECT?
1394	004754	001401				BEQ .+4	;YES
1395	004756	104200				HLT !CS2	;OR SHOULD BE 1
1396	004760	005001				CLR GOOD	;CLEAR LOCATION COUNTER
1397	004762	005201				2\$: INC GOOD	;ADD 1 TO IT
1398	004764	022701	000103			CMP #67.,GOOD	;LAST LOC YET?
1399	004770	001405				BEQ 3\$	;YES
1400	004772	017700	174130			MOV @RSDB,BAD	;GET LOC FROM DB
1401	004776	020100				CMP GOOD,BAD	;DO LOCATIONS MATCH?
1402	005000	001770				BEQ 2\$	;YES
1403	005002	104000				HLT	;CAN NOT MATCH 66 LOCATIONS
1404	005004	032777	000200	174074		3\$: BIT #OR,@RSCS2	;IS OR 0
1405	005012	001401				BEQ .+4	;YES
1406	005014	104200				HLT !CS2	;OR SHOULD BE 0
1407							
1408						;NOW PUT 67 WORDS INTO SILO AND CHECK FOR DLT ERROR	
1409							
1410	005016	005001				4\$: CLR R1	;CLEAR COUNTER
1411	005020	005201				INC R1	;ADD 1 TO COUNT
1412	005022	010177	174100			MOV R1,@RSDB	;PUT INTO COUNTER
1413	005026	022701	000103			CMP #67.,R1	;DONE YET?
1414	005032	001401				BEQ .+4	;YES
1415	005034	000771				BR 4\$	;NO DO AGAIN
1416	005036	032777	100000	174042		BIT #DLT,@RSCS2	;DID DATA LATE SET?
1417	005044	001001				BNE .+4	;YES
1418	005046	104200				HLT !CS2	;DLT DID NOT SET
1419							
1420						;DOES SILO CHANGE WITH 67TH WORD: IT SHOULD NOT	
1421							
1422	005050	017700	174052			MOV @RSDB,BAD	;GET 1ST WD FORM SILO
1423	005054	012701	000001			MOV #1,GOOD	;CORRECT ANS OF SILO
1424	005060	020100				CMP GOOD,BAD	;IS SILO GOOD
1425	005062	001401				BEQ .+4	;YES
1426	005064	104000				HLT	;SILO SHOULD NOT HAVE MOVED
1427	005066	104400				TST40: SCOPE	

CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 F 3 PAGE 32
TST36 LOAD RSDB WITH ALL ONES AND ALL ZEROS

SEQ 0031

```
1428 ;FLOAT A 1 AND A 0 THROUGH THE SILO
1429
1430 SILOFL: CLRDK ;CLEAR ALL RS REG
1431 CLC ;CLEAR CARRY TO FLOAT A 0
1432 MOV #1,GOOD ;GET UP DATA FOR INPUT TO SILO
1433 1$: MOV GOOD,@RSDB ;LOAD DB
1434 ROL GOOD ;SHIFT BIT
1435 BCS .+4 ;DONE YET SHIFTING?
1436 BR 1$ ;NO
1437 MOV #-2,GOOD ;SET ALL ONES
1438 SEC ;SET CARRY TO ROL
1439 3$: MOV GOOD,@RSDB ;LOAD SILO
1440 ROL GOOD ;SHIFT 0
1441 BCS 3$ ;LOOP TILL DONE
1442
1443 ;NOW TEST OUTPUT
1444
1445 CLC ;CLEAR CARRY
1446 MOV #1,GOOD ;CORRECT ANS
1447 2$: MOV @RSDB,BAD ;GET DATA FROM DB
1448 CMP GOOD,BAD ;IS DB DATA GOOD?
1449 BEQ .+4 ;YES
1450 HLT ;DB COULD NOT BUBBLE CORRECTLY
1451 ROL GOOD ;SETUP FOR NEXT ANS
1452 BCS .+4 ;DONE YET?
1453 BR 2$ ;NO
1454 MOV #-2,GOOD ;SETUP FOR ANS
1455 4$: MOV @RSDB,BAD ;GET DATA FROM DB
1456 CMP GOOD,BAD ;IS IT CORRECT?
1457 BEQ .+4 ;YES
1458 HLT ;DB WRONG
1459 SEC ;SET CARRY TO ROL
1460 ROL GOOD ;SETUP FOR NEXT ANS
1461 BCS 4$ ;LOOP TILL DONE
```



```

1462 ;*****
1463 ;TEST 41 TEST NO-OP FUNCTION
1464 ;*****
1465 005202 104400 TST41: SCOPE
1466
1467 005204 104414 NOOP: CLRDK ;CLEAR ALL RS REG
1468 005206 012737 177777 027530 MOV #177777,OUTBUF ;DATA TO BE XFERED
1469 005214 013777 001102 173670 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1470 005222 012777 177777 173660 MOV #-1,@RSWC ;LOAD WC WITH -1
1471 005230 012777 000001 173646 MOV #1,@RSCS1 ;DO NO-OP FUNCTION
1472 005236 032777 000001 173640 BIT #1,@RSCS1 ;DID GO BIT CLEAR
1473 005244 001401 BEQ .+4 ;YES
1474 005246 104001 HLT !CS1 ;GO BIT SHOULD BE CLEARED
1475 005250 005777 173644 TST @RSER ;DID ANY ERRORS OCCUR?
1476 005254 001401 BEQ .+4 ;NO
1477 005256 104002 HLT !ER ;ALL ERROR BITS SHOULD BE 0
1478 005260 022777 177777 173622 CMP #-1,@RSWC ;DID WC MOVE?
1479 005266 001401 BEQ .+4 ;NO
1480 005270 104010 HLT !WC ;WC SHOULD = 1777777
1481 005272 005777 173616 TST @RSDA ;DID DA MOV
1482 005276 001401 BEQ .+4 ;NO
1483 005300 104004 HLT !DA ;DA SHOULD =0
1484 005302 023777 001102 173602 CMP @#OBUFSV,@RSBA ;DID BA MOVE
1485 005310 001401 BEQ .+4 ;NO
1486 005312 104020 HLT !BA ;BA MOVED
1487 005314 033777 001156 173600 BIT UNCMP,@RSAS ;AS SHOULD NOT SET ON
1488 005322 001401 BEQ .+4 ;A NO-OP FUNCTION
1489 005324 104100 HLT !AS ;AS SET WHY?

```



```

1490
1491
1492
1493 005326 104400
1494
1495 005330 104414
1496 005332 012777 000007 173560
1497 005340 033777 001156 173554
1498 005346 001001
1499 005350 104100
1500 005352 012737 177777 027530
1501 005360 013777 001102 173524
1502 005366 012777 177777 173514
1503 005374 012777 000001 173502
1504 005402 032777 000001 173474
1505 005410 001401
1506 005412 104001
1507 005414 022777 150600 173474
1508 005422 001401
1509 005424 104040
1510 005426 022777 177777 173454
1511 005434 001401
1512 005436 104010
1513 005440 005777 173450
1514 005444 001401
1515 005446 104004
1516 005450 023777 001102 173434
1517 005456 001401
1518 005460 104020
1519 005462 033777 001156 173432
1520 005470 001001
1521 005472 104100
1522 005474 022777 000007 173416
1523 005502 001401
1524 005504 104002

```

:TEST 42 TEST NO-OP FUNCTION WITH ERROR BITS SET

TST42: SCOPE

NNOOP: CLRDK ;CLEAR ALL RS REG
MOV #7, @RSER ;LOAD ER
BIT UNCMP, @RSAS ;IS ATA BIT SET?
BNE .+4 ;YES
HLT !AS ;AS BIT SHOULD BE SET
MOV #177777, OUTBUF ;DATA TO BE XFERED
MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
MOV #-1, @RSWC ;LOAD WC WITH -1
MOV #1, @RSCS1 ;DO NO-OP FUNCTION
BIT #1, @RSCS1 ;DID GO BIT CLEAR
BEQ .+4 ;YES
HLT !CS1 ;GO BIT SHOULD BE CLEARED
CMP #150600, @RSDS ;DID ERR BITS SET?
BEQ .+4 ;NO
HLT !DS ;ERR BIT SHOULD BE 0
CMP #-1, @RSWC ;DID WC MOVE?
BEQ .+4 ;NO
HLT !WC ;WC SHOULD = 177777
TST @RSDA ;DID DA MOV
BEQ .+4 ;NO
HLT !DA ;DA SHOULD =0
CMP @#OBUFSV, @RSBA ;DID BA MOVE
BEQ .+4 ;NO
HLT !BA ;BA MOVED
BIT UNCMP, @RSAS ;AS SHOULD BE SET
BNE .+4 ;IS IT?
HLT !AS ;NO
CMP #7, @RSER ;DID ER CHANGE?
BEQ .+4 ;NO
HLT !ER ;ER SHOULD NOT CHANGE


```

1525 ;*****
1526 ;TEST 43 TEST NO-OP FUNCTION CODE 21
1527 ;*****
1528 005506 104400 TST43: SCOPE
1529
1530 NOOP21: CLRDK ;CLEAR ALL RS REG
1531 005510 104414 MOV #177777,OUTBUF ;DATA TO BE XFERED
1532 005512 012737 177777 027530 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1533 005520 013777 001102 173364 MOV #-1,@RSWC ;LOAD WC WITH -1
1534 005526 012777 177777 173354 MOV #21,@RSCS1 ;DO NO-OP FUNCTION
1535 005534 012777 000021 173342 BIT #1,@RSCS1 ;DID GO BIT CLEAR
1536 005542 032777 000001 173334 BEQ .+4 ;YES
1537 005550 001401 HLT !CS1 ;GO BIT SHOULD BE CLEARED
1538 005552 104001 TST @RSER ;DID ANY ERRORS OCCUR?
1539 005554 005777 173340 BEQ .+4 ;NO
1540 005560 001401 HLT !ER ;ALL ERROR BITS SHOULD BE 0
1541 005562 104002 177777 173316 CMP #-1,@RSWC ;DID WC MOVE?
1542 005572 001401 BEQ .+4 ;NO
1543 005574 104010 HLT !WC ;WC SHOULD = 177777
1544 005576 005777 173312 TST @RSDA ;DID DA MOV
1545 005602 001401 BEQ .+4 ;NO
1546 005604 104004 HLT !DA ;DA SHOULD =0
1547 005606 023777 001102 173276 CMP @#OBUFSV,@RSBA ;DID BA MOVE
1548 005614 001401 BEQ .+4 ;NO
1549 005616 104020 HLT !BA ;BA MOVED
1550 005620 033777 001156 173274 BIT UNCMP,@RSAS ;AS SHOULD NOT SET ON
1551 005626 001401 BEQ .+4 ;A NO-OP FUNCTION
1552 005630 104100 HLT !AS ;AS SET WHY?
1553 005632 022777 004220 173244 CMP #4220,@RSCS1 ;IS CS1 CORRECT?
1554 005640 001401 BEQ .+4 ;YES
1555 005642 104040 HLT !DS ;CS1 SHOULD = 4220

```



```

1556 ;*****
1557 ;TEST 44 TEST NO-OP FUNCTION CODE 21 WITH ERROR BITS SET
1558 ;*****
1559 005644 104400 TST44: SCOPE
1560
1561 005646 104414 NNOP21: CLRDK ;CLEAR ALL RS REG
1562 005650 012777 000007 173242 MOV #7,ARSER ;LOAD ER
1563 005656 033777 001156 173236 BIT UNCMP,ARSAS ;IS ATA BIT SET?
1564 005664 001001 BNE .+4 ;YES
1565 005666 104100 HLT !AS ;AS BIT SHOULD BE SET
1566 005670 012737 177777 027530 MOV #177777,OUTBUF ;DATA TO BE XFERED
1567 005676 013777 001102 173206 MOV @#OBUFSV,ARSBA ;SET UP CURRENT ADDRESS
1568 005704 012777 177777 173176 MOV #-1,ARSWC ;LOAD WC WITH -1
1569 005712 012777 000021 173164 MOV #21,ARSCS1 ;DO NO-OP FUNCTION
1570 005720 032777 000001 173156 BIT #1,ARSCS1 ;DID GO BIT CLEAR
1571 005726 001401 BEQ .+4 ;YES
1572 005730 104001 HLT !CS1 ;GO BIT SHOULD BE CLEARED
1573 005732 022777 150600 173156 CMP #150600,ARSDS ;DID ERR BITS SET?
1574 005740 001401 BEQ .+4 ;NO
1575 005742 104040 HLT !DS ;ERR BIT SHOULD BE 0
1576 005744 022777 177777 173136 CMP #-1,ARSWC ;DID WC MOVE?
1577 005752 001401 BEQ .+4 ;NO
1578 005754 104010 HLT !WC ;WC SHOULD = 177777
1579 005756 005777 173132 TST ARSDA ;DID DA MOV
1580 005762 001401 BEQ .+4 ;NO
1581 005764 104004 HLT !DA ;DA SHOULD =0
1582 005766 023777 001102 173116 CMP @#OBUFSV,ARSBA ;DID BA MOVE
1583 005774 001401 BEQ .+4 ;NO
1584 005776 104020 HLT !BA ;BA MOVED
1585 006000 033777 001156 173114 BIT UNCMP,ARSAS ;AS SHOULD BE SET
1586 006006 001001 BNE .+4 ;IS IT?
1587 006010 104100 HLT !AS ;NO
1588 006012 022777 000007 173100 CMP #7,ARSER ;DID ER CHANGE?
1589 006020 001401 BEQ .+4 ;NO
1590 006022 104002 HLT !ER ;ER SHOULD NOT CHANGE
1591 006024 022777 104220 173052 CMP #104220,ARSCS1 ;IS CS1 CORRECT?
1592 006032 001401 BEQ .+4 ;YES
1593 006034 104040 HLT !DS ;CS1 SHOULD = 104220

```



```

1594
1595
1596
1597 006036 104400
1598
1599
1600
1601 006040 104414
1602 006042 012777 177777 173040
1603 006050 012777 177777 173036
1604 006056 012777 177017 173034
1605 006064 012777 000070 173036
1606 006072 012777 000011 173004
1607 006100 017700 173002
1608 006104 042700 177640
1609 006110 013701 001152
1610 006114 052701 000100
1611 006120 020100
1612 006122 001401
1613 006124 104000
1614 006126 005777 172762
1615 006132 001401
1616 006134 104004
1617 006136 005777 172756
1618 006142 001401
1619 006144 104002
1620 006146 017737 172756 001176
1621 006154 042737 177707 001176
1622 006162 022737 000070 001176
1623 006170 001401
1624 006172 104220
1625 006174 022777 004210 172702
1626 006202 001401
1627 006204 104001
1628 006206 033777 001156 172706
1629 006214 001401
1630 006216 104100
1631 006220 022777 177777 172662
1632 006226 001401
1633 006230 104010

```

```

*****
:TEST 45 TEST DRIVE CLEAR FUNCTION WITH ERRORS SET
*****
TST45: SCOPE
:FIRST SET ALL R/W BITS IN DISK REG
:DO DRIVE CLEAR-ALL R/W BITS SHOULD BE CLEARED

DRCLR: CLRDK
MOV #177777,@RSWC ;CLEAR ALL RS REG
MOV #177777,@RSDA ;LOAD RSWC
MOV #177017,@RSER ;SET ALL POSSIBLE
MOV #70,@RSMR ;BITS IN DISK REG
MOV #11,@RSCS1 ;SET THESE BITS
MOV @RSCS2,BAD ;SET DRIVE CLEAR
MOV #177640,BAD ;GET CS2 DATA
BIC UNNUM,GOOD ;CLEAR JUNK
MOV #100,GOOD ;GET DRIVE UNIT
BIS GOOD,BAD ;SET IR BIT
CMP .+4 ;IS UNIT # THE SAME
BEQ .+4 ;YES
HLT ;UNIT # IN CS2 GOT MODIFIED
TST @RSDA ;DID DA CLEAR
BEQ .+4 ;YES
HLT !DA ;DA SHOULD BE 0
TST @RSER ;DID ER CLEAR
BEQ .+4 ;YES
HLT !ER ;ER SHOULD BE CLEARED
MOV @RSMR,WORK ;GET MR REG
BIC #177707,WORK ;CLEAR JUNK
CMP #70,WORK ;IS 70 STILL SET IN MR?
BEQ .+4 ;YES
HLT !MR ;BITS 70 SHOULD NOT CLEAR
CMP #4210,@RSCS1 ;DID THESE BITS CLEAR?
BEQ .+4 ;YES
HLT !CS1 ;CS1 SHOULD =4210
BIT UNCMP,@RSAS ;AS SHOULD NOT SET
BEQ .+4 ;ON A DRIVE CLEAR FUN
HLT !AS ;WHY DID AS SET?
CMP #177777,@RSWC ;DID RSWC CHANGE?
BEQ .+4 ;NO
HLT !WC

```



```
1634 ;DO ONE WORD WRITE
1635 ;*****
1636 ;TEST 46 EXECUTE THE ONE WORD WRITE
1637 ;*****
1638 006232 104400 TST46: SCOPE
1639
1640 006234 104414 WRTST: CLRDK ;CLEAR ALL RS REG
1641 006236 012737 177777 027530 MOV #177777,OUTBUF ;DATA TO BE X-FERED
1642 006244 013777 001102 172640 MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
1643 006252 012777 177777 172630 MOV #-1, @RSWC ;SET WORD COUNT TO -1
1644 006260 012777 000060 172616 1$: MOV #60, @RSCS1 ;SET FUNCTION WITH NO GO BIT
1645 006266 022777 177777 172614 CMP #-1, @RSWC ;DID WC MOVE?
1646 006274 001401 BEQ .+4 ;NO
1647 006276 104010 HLT !WC ;WC MOVED
1648 006300 023777 001102 172604 CMP @#OBUFSV, @RSBA ;DID RSBA MOVE?
1649 006306 001401 BEQ .+4 ;NO
1650 006310 104020 HLT !BA ;BA MOVED
1651 006312 052777 000001 172564 BIS #BIT0, @RSCS1 ;SET GO BIT
1652 006320 105777 172560 2$: TSTB @RSCS1 ;TEST FOR RDY=0
1653 006324 100001 BPL .+4 ;RDY=0
1654 006326 104001 HLT !CS1 ;RDY SHOULD = 0
1655 006330 004737 026744 JSR PC, WAITRY ;WAIT FOR READY
1656 006334 104001 HLT !CS1 ;SHOULD = 260 RDY NEVER CAME UP
1657 006336 022777 000001 172550 CMP #1, @RSDA ;IS RSDA CORRECT
1658 006344 001401 BEQ .+4 ;RSDA OK
1659 006346 104004 HLT !DA ;SHOULD = 1 SHOULD INCREMENT
1660 006350 022777 004260 172526 3$: CMP #4260, @RSCS1 ;IS ERROR FLAG SET?
1661 006356 001401 BEQ .+4 ;NO! X-FER OK
1662 006360 104047 HLT !CS1!ER!DS!DA ;ERROR DURING X-FER
1663 006362 005777 172522 4$: TST @RSWC ;FETCH WORD COUNT
1664 006366 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1665 006370 104010 HLT !WC ;SHOULD = 0 FAILED TO INCREMENT
1666 006372 022777 010600 172516 CMP #10600, @RSDS ;IS RSDS OK?
1667 006400 001401 BEQ .+4 ;YES
1668 006402 104044 HLT !DS!DA ;NO
1669 006404 013701 001152 MOV UNNUM, GOOD ;GET UNIT #
1670 006410 052701 000100 BIS #100, GOOD ;SET IR BIT
1671 006414 017700 172466 MOV @RSCS2, BAD ;GET CS2
1672 006420 020100 CMP GOOD, BAD ;IS CS2 CORRECT?
1673 006422 001401 BEQ .+4 ;YES
1674 006424 104000 HLT ;BAD = CS2 GOOD IS CORRECT ANS
1675 006426 017700 172460 MOV @RSBA, BAD ;GET BA DATA
1676 006432 013701 001102 MOV @#OBUFSV, GOOD ;WHAT RSBA SHOULD EQUAL
1677 006436 062701 000002 ADD #2, GOOD ;UPDATE OUTBUFFER
1678 006442 020100 CMP GOOD, BAD ;IS RSBA CORRECT
1679 006444 001401 BEQ .+4 ;YES
1680 006446 104000 HLT ;BA FAILED TO INCREMENT
1681 006450 005777 172444 TST @RSER ;DID ANY ERRORS SET?
1682 006454 001401 BEQ .+4 ;NO
1683 006456 104040 HLT !DS
```



```

1684 ;TEST READ FUNCTION
1685
1686 ;*****
1687 ;TEST 47 EXECUTE THE ONE WORD READ
1688 ;*****
1689 006460 104400 TST47: SCOPE
1690
1691 006462 104414 RDTST: CLRDK ;CLEAR ALL RS REG
1692 006464 005037 027530 CLR ;CLR TO READ INTO
1693 006470 013777 001102 172414 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1694 006476 012777 177777 172404 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1695 006504 012777 000071 172372 1$: MOV #71,@RSCS1 ;GO READ
1696 006512 105777 172366 2$: TSTB @RSCS1 ;TEST FOR BUSY=1
1697 006516 100001 BPL .+4 ;BUSY SET
1698 006520 104001 HLT !CS1 ;BUSY NOT SET
1699 006522 004737 026744 JSR PC,WAITRY ;WAIT FOR READY
1700 006526 104001 HLT !CS1 ;TIMEOUT RDY DID NOT SET
1701 006530 022777 000001 172356 CMP #BIT0,@RSDA ;WAS RSDA INCREMENTED BY 1
1702 006536 001401 BEQ .+4 ;RSDA OK
1703 006540 104046 HLT !DA!ER!DS ;RSDA SHOULD CONTAIN A 1
1704 006542 022777 004270 172334 3$: CMP #4270,@RSCS1 ;IS ERROR FLAG SET?
1705 006550 001401 BEQ .+4 ;NO! X-FER OK
1706 006552 104043 HLT !CS1!ER!DS ;RSCS1 SHOULD = 270
1707 006554 005777 172330 4$: TST @RSWC ;TEST WC
1708 006560 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1709 006562 104010 HLT !WC ;SHOULD = 0
1710 006564 013701 001152 MOV UNNUM,GOOD ;GET CORRECT
1711 006570 052701 000100 BIS #100,GOOD ;ANS OF CS2
1712 006574 017700 172306 MOV @RSCS2,BAD ;GET CS2
1713 006600 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1714 006602 001401 BEQ .+4 ;YES
1715 006604 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1716 006606 017700 172300 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1717 006612 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1718 006616 062701 000002 ADD #2,GOOD ;UPDATE IT
1719 006622 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1720 006624 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1721 006626 104000 HLT ;RSBA FAILED TO INCREMENT
1722 006630 013700 027530 MOV OUTBUF,BAD ;GET DATA READ FROM DISK
1723 006634 012701 177777 MOV #-1,GOOD ;GET CORRECT ANS
1724 006640 020100 CMP GOOD,BAD ;IS OUTBUF CORRECT
1725 006642 001401 BEQ .+4 ;YES
1726 006644 104000 HLT ;GOOD=CORRECT ANS BAD=DATA READ FROM DISK

```



```

1727 ;*****
1728 ;TEST 50 TEST WRITE CHECK
1729 ;*****
1730 006646 104400 TST50: SCOPE
1731 ;DO A ONE WORD WRITE CHECK
1732
1733 ;* * *EXECUTE THE ONE WORD WRITE CHECK* * *
1734
1735 006650 104414 WRCKT: CLRDK ;CLEAR ALL RS REG
1736 006652 012737 177777 027530 MOV #177777,OUTBUF ;DATA TO BE X-FERED
1737 006660 013777 001102 172224 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1738 006666 012777 177777 172214 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1739 006674 012777 000051 172202 1$: MOV #51,@RSCS1 ;GO WRITE CHECK
1740 006702 105777 172176 2$: TSTB @RSCS1 ;TEST FOR READY
1741 006706 100001 BPL .+4 ;NOT READY
1742 006710 104001 HLT !CS1 ;BUSY FAILED TO SET
1743 006712 004737 026744 RSWCWT: JSR PC,WAITRY ;WAIT FOR READY
1744 006716 104001 HLT !CS1 ;BUSY FAILED TO CLEAR
1745 006720 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1746 006724 052701 000100 BIS #100,GOOD ;SET BIT IR
1747 006730 017700 172152 MOV @RSCS2,BAD ;GET CS2
1748 006734 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1749 006736 001401 BEQ .+4 ;YES
1750 006740 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1751 006742 022777 004250 172134 2$: CMP #4250,@RSCS1 ;ANY ERRORS?
1752 006750 001401 BEQ .+4 ;X-FER OK
1753 006752 104046 HLT !DA!ER!DS ;ERROR DUR X-FER
1754 006754 022777 000001 172132 3$: CMP #BIT0,@RSDA ;WAS DAR INCREMENTED BY 1
1755 006762 001401 BEQ .+4 ;RSDA OK
1756 006764 104004 HLT !DA ;DAR SHOULD = 1
1757 006766 005777 172116 TST @RSWC ;TEST FOR OVERFLOW
1758 006772 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1759 006774 104010 HLT !WC ;SHOULD = 0
1760 006776 017700 172110 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1761 007002 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1762 007006 062701 000002 ADD #2,GOOD ;UPDATE IT
1763 007012 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1764 007014 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1765 007016 104000 HLT ;RSBA FAILED TO INCREMENT

```



```
1766      ;DO ONE WORD WRITE ON -B- PORT
1767      ;IF A 1 WD TRANSFER KEEPS SETTING NEM PROGAM WILL GO AND UPDATE
1768      ;ADDRESS (OBUFSV) ON -B- PORT BY 4K AND TRY TRANSFER AGAIN UNTILL IT
1769      ;REACHES 28K. IF NO TRANSFER IT THEN SKIPS WRITE,
1770      ;READ AND WRITE CHECK ON -B- PORT
1771      ;TO INHIBIT OBUFSV FROM CHANGING SET BIT 12
1772
1773      ;*****
1774      ;TEST 51 EXECUTE THE ONE WORD WRITE ON -B- PORT
1775      ;*****
1776 007020 104400 TST51: SCOPE
1777
1778 007022 104414 WRTSTB: CLRDK      ;CLEAR ALL RS REG
1779 007024 013737 001102 001176 MOV @#OBUFSV,WORK ;GET LOC OF RSBA TO LOAD
1780 007032 012777 177777 172136 MOV #177777,@WORK ;DATA TO BE X-FERED
1781 007040 013777 001102 172044 MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
1782 007046 012777 177777 172034 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1783 007054 012777 002061 172022 MOV #2061,@RSCS1 ;TEST B PORT
1784 007062 105777 172016 2$: TSTB @RSCS1 ;TEST FOR RDY=0
1785 007066 100001 BPL .+4 ;RDY=0
1786 007070 104001 HLT !CS1 ;RDY SHOULD = 0
1787 007072 004737 026744 JSR PC, WAITRY ;WAIT FOR READY
1788 007076 104001 HLT !CS1 ;SHOULD = 260 RDY NEVER CAME UP
1789 007100 032777 010000 171720 BIT #BIT12,@SWR ;INHIBIT ADDRESS?
1790 007106 001006 BNE 3$ ;YES
1791 007110 032777 004000 171770 BIT #BIT11,@RSCS2 ;DID NEM SET?
1792 007116 001402 BEQ 3$ ;NO
1793 007120 000137 025776 JMP FINDM ;GO FIND MEMORY ON PORT B
1794 007124 022777 000001 171762 3$: CMP #1,@RSDA ;IS RSDA CORRECT
1795 007132 001401 BEQ .+4 ;RSDA OK
1796 007134 104004 HLT !DA ;SHOULD = 1 SHOULD INCREMENT
1797 007136 022777 006260 171740 CMP #6260,@RSCS1 ;IS CS1 CORRECT?
1798 007144 001401 BEQ 4$ ;YES
1799 007146 104001 HLT !CS1
1800 007150 005777 171734 4$: TST @RSWC ;FETCH WORD COUNT
1801 007154 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1802 007156 104010 HLT !WC ;SHOULD = 0 FAILED TO INCREMENT
1803 007160 022777 010600 171730 CMP #10600,@RSDS ;IS RSDS OK?
1804 007166 001401 BEQ .+4 ;YES
1805 007170 104044 HLT !DS!DA ;NO
1806 007172 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1807 007176 052701 000100 BIS #100,GOOD ;SET IR BIT
1808 007202 017700 171700 MOV @RSCS2,BAD ;GET CS2
1809 007206 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1810 007210 001401 BEQ .+4 ;YES
1811 007212 104000 HLT ;BAD = CS2 GOOD IS CORRECT ANS
1812 007214 017700 171672 MOV @RSBA,BAD ;GET BA DATA
1813 007220 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1814 007224 062701 000002 ADD #2,GOOD ;UPDATE OUTBUFFER
1815 007230 020100 CMP GOOD,BAD ;IS RSBA CORRECT
1816 007232 001401 BEQ .+4 ;YES
1817 007234 104000 HLT ;BA FAILED TO INCREMENT
```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 42
TST52 EXECUTE THE ONE WORD READ ON -B- PORT

SEQ 0041

```
1818 ;*****
1819 ;TEST 52 EXECUTE THE ONE WORD READ ON -B- PORT
1820 ;*****
1821 007236 104400 TST52: SCOPE
1822
1823 007240 104414 RDTSTB: CLRDK ;CLEAR ALL RS REG
1824 007242 005037 001176 CLR ;CLR TO READ INTO
1825 007246 013777 001102 171636 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1826 007254 012777 177777 171626 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1827 007262 012777 002071 171614 MOV #2071,@RSCS1 ;B PORT
1828 007270 105777 171610 2$: TSTB @RSCS1 ;TEST FOR BUSY=1
1829 007274 100001 BPL .+4 ;BUSY SET
1830 007276 104001 HLT !CS1 ;BUSY NOT SET
1831 007300 004737 026744 JSR PC,WAITRY ;WAIT FOR READY
1832 007304 104001 HLT !CS1 ;TIMEOUT RDY DID NOT SET
1833 007306 022777 000001 171600 CMP #BIT0,@RSDA ;WAS RSDA INCREMENTED BY 1
1834 007314 001401 BEQ .+4 ;RSDA OK
1835 007316 104046 HLT !DA!ER!DS ;RSDA SHOULD CONTAIN A 1
1836 007320 022777 006270 171556 CMP #6270,@RSCS1 ;TST B PORT
1837 007326 001401 BEQ 4$ ;OK
1838 007330 104001 HLT !CS1 ;CS1 SHOULD = 6270
1839 007332 005777 171552 4$: TST @RSWC ;TEST WC
1840 007336 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1841 007340 104010 HLT !WC ;SHOULD = 0
1842 007342 013701 001152 MOV UNNUM,GOOD ;GET CORRECT
1843 007346 052701 000100 BIS #100,GOOD ;ANS OF CS2
1844 007352 017700 171530 MOV @RSCS2,BAD ;GET CS2
1845 007356 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1846 007360 001401 BEQ .+4 ;YES
1847 007362 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1848 007364 017700 171522 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1849 007370 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1850 007374 062701 000002 ADD #2,GOOD ;UPDATE IT
1851 007400 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1852 007402 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1853 007404 104000 HLT ;RSBA FAILED TO INCREMENT
1854 007406 013737 001102 001176 MOV @#OBUFSV,WORK ;GET DATA READ FROM DISK
1855 007414 017700 171556 MOV @WORK,BAD
1856 007420 012701 177777 MOV #-1,GOOD ;GET CORRECT ANS
1857 007424 020100 CMP GOOD,BAD ;IS OUTBUF CORRECT
1858 007426 001401 BEQ .+4 ;YES
1859 007430 104000 HLT ;GOOD=CORRECT ANS BAD=DATA READ FROM DISK
```



```
1860 ;*****
1861 ;TEST 53 TEST WRITE CHECK ON -B- PORT
1862 ;*****
1863 007432 104400 TST53: SCOPE
1864
1865
1866 007434 104414 WRCKTB: CLRDK ;CLEAR ALL RS REG
1867 007436 013737 001102 001176 MOV @#OBUFSV,WORK ;GET LOC FOR
1868 007444 012777 177777 171524 MOV #177777,@WORK ;DATA TO BE X-FERED
1869 007452 013777 001102 171432 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1870 007460 012777 177777 171422 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1871 007466 012777 002051 171410 MOV #2051,@RSCS1 ;B PORT
1872 007474 105777 171404 2$: TSTB @RSCS1 ;TEST FOR READY
1873 007500 100001 BPL .+4 ;NOT READY
1874 007502 104001 HLT !CS1 ;BUSY FAILED TO SET
1875 007504 004737 026744 JSR PC,WAITRY ;WAIT FOR READY
1876 007510 104001 HLT !CS1 ;BUSY FAILED TO CLEAR
1877 007512 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1878 007516 052701 000100 BIS #100,GOOD ;SET BIT IR
1879 007522 017700 171360 MOV @RSCS2,BAD ;GET CS2
1880 007526 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1881 007530 001401 BEQ .+4 ;YES
1882 007532 104000 HLT ;GOOD = CORRECT ANS FOR CS2
1883 007534 022777 006250 171342 CMP #6250,@RSCS1 ;IS CS1 CORRECT?
1884 007542 001401 BEQ 3$ ;YES
1885 007544 104001 HLT !CS1 ;CS1 SHOULD = 6250
1886 007546 022777 000001 171340 3$: CMP #BIT0,@RSDA ;WAS DAR INCREMENTED BY 1
1887 007554 001401 BEQ .+4 ;RSDA OK
1888 007556 104004 HLT !DA ;DAR SHOULD = 1
1889 007560 005777 171324 TST @RSWC ;TEST FOR OVERFLOW
1890 007564 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1891 007566 104010 HLT !WC ;SHOULD = 0
1892 007570 017700 171316 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
1893 007574 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1894 007600 062701 000002 ADD #2,GOOD ;UPDATE IT
1895 007604 020001 CMP BAD,GOOD ;IS RSBA CORRECT
1896 007606 001401 BEQ .+4 ;YES EXECUTE CONTINUE
1897 007610 104000 HLT ;RSBA FAILED TO INCREMENT
```



```
1898 007612 013737 001102 001170 NXM: MOV OBUFSV,BPORTT ;SAVE -B- PORT BUFFER
1899 007620 012737 027530 001102 MOV #OUTBUF,OBUFSV ;RESTORE OBUFSV
1900
1901 ;DESELECT THEN SELECT UNIT NUMBER IN RSCS2 CHECK TIMING
1902 ;*****
1903 ;TST 54 DESELECT THEN SELECT UNIT NUMBER TIMING TEST
1904 ;*****
1905 007626 104400 TST54: SCOPE
1906
1907 007630 104414 UNITST: CLRDK ;CLEAR ALL RS REG
1908 007632 005004 CLR R4 ;CLEAR R4
1909 007634 020437 001152 CMP R4,UNNUM ;IS THIS CORRECT UNIT #?
1910 007640 001001 BNE 3$ ;NO THEN USE IT
1911 007642 005204 INC R4 ;GET WRONG DRIVE
1912 007644 012737 177777 027530 3$: MOV #177777,OUTBUF ;DATA TO BE X-FERED
1913 007652 013777 001102 171232 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
1914 007660 012777 177777 171222 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1915 007666 012703 000061 MOV #61,R3 ;GET WRITE FUNCTION
1916 007672 013705 001152 MOV UNNUM,R5 ;GET CORRECT UNIT #
1917 007676 012701 172040 MOV #172040,R1 ;GET CS1 REG
1918 007702 010461 000010 MOV R4,10(R1) ;LOAD WRONG UNIT # INTO CS2
1919 007706 000240 NOP ;WAIT FOR DRIVE TO SETTLE
1920 007710 010561 000010 MOV R5,10(R1) ;LOAD CORRECT UNIT #
1921 007714 010311 1$: MOV R3,(R1) ;LOAD FUNCTION IN CS1
1922 007716 004737 026744 JSR PC,WAITRY ;WAIT FOR READY
1923 007722 104001 HLT !CS1 ;SHOULD = 260 RDY NEVER CAME UP
1924 007724 022777 004260 171152 CMP #4260,@RSCS1 ;IS ERROR FLAG SET?
1925 007732 001401 BEQ .+4 ;NO! X-FER OK
1926 007734 104047 HLT !CS1!ER!DS!DA ;ERROR DURING X-FER
1927 007736 022777 000001 171150 CMP #1,@RSDA ;IS RSDA CORRECT
1928 007744 001401 BEQ .+4 ;RSDA OK
1929 007746 104004 HLT !DA ;SHOULD = 1 SHOULD INCREMENT
1930 007750 005777 171134 TST @RSWC ;FETCH WORD COUNT
1931 007754 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
1932 007756 104010 HLT !WC ;SHOULD = 0 FAILED TO INCREMENT
1933 007760 022777 010600 171130 CMP #10600,@RSDS ;IS RSDS OK?
1934 007766 001401 BEQ .+4 ;YES
1935 007770 104044 HLT !DS!DA ;NO
1936 007772 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
1937 007776 052701 000100 BIS #100,GOOD ;SET IR BIT
1938 010002 017700 171100 MOV @RSCS2,BAD ;GET CS2
1939 010006 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
1940 010010 001401 BEQ .+4 ;YES
1941 010012 104000 HLT ;BAD = CS2 GOOD IS CORRECT ANS
1942 010014 017700 171072 MOV @RSBA,BAD ;GET BA DATA
1943 010020 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
1944 010024 062701 000002 ADD #2,GOOD ;UPDATE OUTBUFFER
1945 010030 020100 CMP GOOD,BAD ;IS RSBA CORRECT
1946 010032 001401 BEQ .+4 ;YES
1947 010034 104000 HLT ;BA FAILED TO INCREMENT
```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 45
TST54 DESELECT THEN SELECT UNIT NUMBER TIMING TEST

SEQ 0044

```
1948 ;TEST CURRENT ADDRESS INHIBT-BAI IN RSCS2
1949 ;DO A ONE WORD WRITE AND SEE
1950 ;IF RSBA INCREMENTED AFTER THE X-FER
1951
1952 ;*****
1953 ;TEST 55 TEST BAI IN RSCS2
1954 ;*****
1955 TST55: SCOPE
1956
1957 010036 104400 BAITST: CLRDK ;CLEAR ALL RS REG
1958 010040 104414 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDR
1959 010042 013777 001102 171042 MOV #-1,@RSWC ;SET WORD COUNT TO -1
1960 010050 012777 177777 171032 MOV #BAI,@RSCS2 ;SET BAI BIT
1961 010056 052777 000010 171022 BIS #61,@RSCS1 ;WRITE
1962 010064 012777 000061 171012 MOV PC,WAITRY ;WAIT FOR READY
1963 010072 004737 026744 JSR !CS1 ;RDY DID NOT SET
1964 010076 104001 HLT ;WHAT RSBA SHOULD BE
1965 010100 013701 001102 1$: MOV @#OBUFSV,GOOD ;WHAT RSBA IS
1966 010104 017700 171002 MOV @RSBA,BAD ;COMPARE
1967 010110 020100 CMP GOOD,BAD ;YES
1968 010112 001401 BEQ .+4 ;BAD=OUTBUF GOOD = CORRECT ANS
1969 010114 104000 HLT ;ANY ERRORS?
1970 010116 005777 170776 TST @RSER ;NO
1971 010122 001401 BEQ .+4 ;YES
1972 010124 104040 HLT !DS ;YES
1973 010126 104414 CLRDK ;CLEAR ALL RS REG
1974 010130 032777 000010 170750 BIT #BAI,@RSCS2 ;DID BAI CLEAR?
1975 010136 001401 BEQ .+4 ;YES
1976 010140 104002 HLT !ER ;BAI DID NOT SET
```



```

1976
1977
1978
1979 010142 104400
1980
1981 010144 104414
1982 010146 052777 000010 170732
1983 010154 012777 177600 170726
1984 010162 012777 173000 170722
1985 010170 012777 001471 170706
1986 010176 004737 026744
1987 010202 104040
1988 010204 013701 001152
1989 010210 052701 004310
1990 010214 017700 170666
1991 010220 020100
1992 010222 001401
1993 010224 104000
1994 010226 022777 145670 170650
1995 010234 001401
1996 010236 104001
1997 010240 012777 040000 170636
1998 010246 017700 170634
1999 010252 013701 001152
2000 010256 052701 000310
2001 010262 020100
2002 010264 001401
2003 010266 104200
2004
2005
2006
2007
2008 010270 104400
2009
2010 010272 104414
2011 010274 012777 000032 170612
2012 010302 013777 001102 170602
2013 010310 012777 177777 170572
2014 010316 005037 001176
2015 010322 032777 001000 170600
2016 010330 001004
2017 010332 005337 001176
2018 010336 001371
2019 010340 104220

;*****
;TEST 56 TEST NON-EXISTENT MEMORY ERROR BIT IN CS2
;*****
TST56: SCOPE

NXMTSM: CLRDK ;CLEAR ALL RS REG
BIS #BAI,@RSCS2 ;SET BAI BIT
MOV #-200,@RSWC ;SET UP WORD COUNT
MOV #173000,@RSBA ;SET UP CURRENT ADDRESS
MOV #1471,@RSCS1 ;READ AND LOAD A16 +A17 FOR 18 BIT ADDRESS
JSR PC,WAITRY ;WAIT FOR READY
HLT !DS ;READY NEVER CAME UP
TSTNEM: MOV UNNUM,GOOD ;GET UNIT NO.
BIS #4310,GOOD ;SET BAI+OR BITS
MOV @RSCS2,BAD ;GET CS2
CMP GOOD,BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT ;BAD=CS2 GOOD=CORRECT ANS FOR CS2
CMP #145670,@RSCS1 ;DID TRE SET?
BEQ .+4 ;YES
HLT !CS1 ;TRE SHOULD SET BECAUSE OF NEM
MOV #TRE,@RSCS1 ;CLEAR TRE
MOV @RSCS2,BAD ;GET CS2
MOV UNNUM,GOOD ;GET DRIVE
BIS #310,GOOD ;SET IR
CMP GOOD,BAD ;IS CS2 CORRECT?
BEQ .+4 ;YES
HLT !CS2 ;CS2=BAD GOOD IS CORRECT ANS FOR CS2

;*****
;TEST 57 TEST BLOCK SEARCH FUNCTION, PIP AND DRY BIT AND ADDR. CONF BIT
;*****
TST57: SCOPE

BLOCK: CLRDK ;CLEAR ALL RS REG
MOV #32,@RSDA ;DO A SEARCH FOR SECTOR 32
MOV @#OBUFSV,@RSBA ;LOAD REGS. TO MAKE
MOV #-1,@RSWC ;SURE THEY DO NOT CHANGE
CLR WORK ;SETUP FOR TIMEOUT ROUTINE
4$: BIT #1000,@RSMR ;WAIT FOR DISK TO
BNE 3$ ;REACH SECTOR 32
DEC WORK ;TIME OUT
BNE 4$ ;ROUTINE
HLT !MR ;COULD NOT FIND SECTOR 32

```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052)
TST57

13-OCT-80 15:16 PAGE 47

TEST BLOCK SEARCH FUNCTION, PIP AND DRY BIT AND ADDR. CONF BIT

SEQ 0046

```
2020 010342 005077 170546 3$: CLR @RSDA ;NOW SEARCH FOR 0
2021 010346 012777 000031 170530 MOV #31,@RSCS1 ;DO A BLOCK SEARCH FUNCTION
2022 010354 032777 000200 170534 BIT #DRY,@RSDS ;IS DRY CLEARED?
2023 010362 001402 BEQ 1$ ;YES
2024 010364 104040 HLT !DS ;DRY SHOULD BE CLEARED DURING A BLOCK SEARCH
2025 010366 000500 BR OOUT ;GET OUT BECAUSE OF TIMING
2026 010370 032777 020000 170520 1$: BIT #20000,@RSDS ;IS PIP SET?
2027 010376 001001 BNE .+4 ;YES
2028 010400 104040 HLT !DS ;PIP SHOULD BE SET
2029 010402 012701 020000 MOV #20000,GOOD ;SETUP FOR TIMEOUT ROUTINE
2030 010406 005301 2$: DEC GOOD ;DO TIMEOUT
2031 010410 001466 BEQ TTMOU ;TIMED OUT
2032 010412 032777 000200 170476 BIT #DRY,@RSDS ;DID DRY SET?
2033 010420 001772 BEQ 2$ ;NO
2034 010422 022777 110600 170466 CMP #110600,@RSDS ;DID PIP CLEAR?
2035 010430 001401 BEQ .+4 ;YES
2036 010432 104040 HLT !DS ;PIP BIT DID NOT CLEAR
2037 010434 022777 104230 170442 CMP #104230,@RSCS1 ;DID SC SET?
2038 010442 001401 BEQ .+4 ;YES
2039 010444 104041 HLT !CS1!DS ;SC DID NOT SET
2040 010446 013737 001152 001176 MOV UNNUM,WORK ;GET CORRECT AS BIT
2041 010454 005001 CLR GOOD ;IN RSAS REG
2042 010456 000261 SEC ;THAT SHOULD
2043 010460 006101 5$: ROL GOOD ;BE SET
2044 010462 005737 001176 TST WORK
2045 010466 001403 BEQ 6$
2046 010470 005337 001176 DEC WORK
2047 010474 000771 BR 5$
2048 010476 020177 170420 6$: CMP GOOD,@RSAS ;IS RSAS CORRECT?
2049 010502 001403 BEQ 7$ ;YES
2050 010504 017700 170412 MOV @RSAS,BAD ;NO
2051 010510 104000 HLT
2052 010512 010177 170404 7$: MOV GOOD,@RSAS ;CLEAR AS REG
2053 010516 005777 170400 TST @RSAS ;DID IT CLEAR?
2054 010522 001401 BEQ .+4 ;YES
2055 010524 104100 HLT !AS ;NO
2056 010526 022777 010600 170362 CMP #10600,@RSDS ;DID ATA CLEAR?
2057 010534 001401 BEQ .+4 ;YES
2058 010536 104040 HLT !DS ;NO
2059 010540 023777 001102 170344 CMP @#OBUFSV,@RSBA ;DID BA MOVE?
2060 010546 001401 BEQ .+4 ;NO
2061 010550 104021 HLT !CS1!BA ;BA MOVED WHY?
2062 010552 022777 177777 170330 CMP #-1,@RSWC ;DID WC MOVE?
2063 010560 001401 BEQ .+4 ;NO
2064 010562 104010 HLT !WC ;WC MOVED WHY?
2065 010564 000401 BR OOUT ;DONE GET OUT
2066 010566 104040 TTMOUT: HLT !DS ;DYR NEVER CAME UP
2067 010570 OOUT: ;DONE
```



```

2068 ;*****
2069 ;TEST 60 ILLEGAL FUNCTION CODE TEST CODE 3 TO 51
2070 ;*****
2071 010570 104400 TST60: SCOPE
2072
2073 ;TEST ILF BIT IN RSER AND ERR BIT IN RSDS
2074 ;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
2075 ;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
2076 ;FUNCTION INTO LOCATION ILLTAB: AND 0 IN FOLLOWING LOCATION
2077
2078 010572 013737 022560 001164 ILL51: MOV TIMES,TIMSV ;SAVE LOOP COUNT
2079 010600 012737 000010 022560 MOV #10,TIMES ;LOOP TEN TIMES
2080 010606 104414 CLRDK ;CLEAR ALL RS REG
2081 010610 012703 027444 1$: MOV #ILLTAB,R3 ;GET STARTING ADD OF TABLE
2082 010614 012300 3$: MOV (R3)+,BAD ;GET ILL FUN
2083 010616 001513 BEQ ILFDN ;DONE GET OUT
2084 010620 013777 001102 170264 MOV @#OBUFSV,@RSBA ;SET UP REGS.
2085 010626 012777 177777 170254 MOV #-1,@RSWC ;TO CHECK FOR CHANGE
2086 010634 010077 170244 2$: MOV BAD,@RSCS1 ;DO ILLEGAL FUNCTION
2087 010640 042700 000001 BIC #BIT0,BAD ;CLEAR GO BIT
2088 010644 010001 MOV BAD,GOOD ;MOV ILLEGAL FUN INTO GOOD
2089 010646 105777 170232 6$: TSTB @RSCS1 ;RDY SET?
2090 010652 100375 BPL 6$ ;NO
2091 010654 052701 104200 BIS #104200,GOOD ;SET ERROR BITS
2092 010660 017700 170220 4$: MOV @RSCS1,BAD ;PUT CS1 INTO BAD
2093 010664 020100 CMP GOOD,BAD ;IS CS1 CORRECT?
2094 010666 001401 BEQ .+4 ;YES
2095 010670 104000 HLT ;GOOD IS WHAT CS1 SHOULD =BAD=CS1
2096 010672 022777 000001 170220 CMP #1,@RSER ;DID ILF SET?
2097 010700 001401 BEQ .+4 ;YES
2098 010702 104043 HLT !CS1!ER!DS ;ILF DID NOT SET
2099 010704 022777 150600 170204 CMP #150600,@RSDS ;IS DS GOOD?
2100 010712 001401 BEQ .+4 ;YES
2101 010714 104043 HLT !CS1!ER!DS ;ERR DID NOT SET
2102 010716 017700 170164 MOV @RSCS2,BAD ;GET CS2
2103 010722 013701 001152 MOV UNNUM,GOOD ;GET UNIT #
2104 010726 052701 000100 BIS #100,GOOD ;SET IR BIT
2105 010732 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2106 010734 001401 BEQ .+4 ;YES
2107 010736 104000 HLT ;GOOD = CORRECT ANS FOR CS2
2108 010740 013701 001156 MOV UNCMP,GOOD ;GET CORRECT DRIVE
2109 010744 042701 177400 BIC #177400,GOOD ;CLEAR UNWANTED BITS
2110 010750 017700 170146 MOV @RSAS,BAD ;GET RSAS REG
2111 010754 020001 CMP BAD,GOOD ;DID CORRECT UNIT ANSWER?
2112 010756 001401 BEQ .+4 ;YES
2113 010760 104100 HLT !AS ;NO WRONG DRIVE ANSWERED

```



```
2114 010762 023777 00.102 170122      CMP      @#OBUFSV,@RSBA      ;DID BA MOVE
2115 010770 001401                      BEQ      .+4                  ;NO
2116 010772 104021                      HLT      !CS1!BA              ;BA MOVED ON AN ILLEGAL FUNCTION
2117 010774 022777 177777 170106      CMP      #-1,@RSWC          ;DID WC MOVE?
2118 011002 001401                      BEQ      .+4                  ;NO
2119 011004 104011                      HLT      !CS1!WC              ;WC MOVED
2120 011006 005777 170102      TST      @RSDA                  ;DID DA MOVE
2121 011012 001401                      BEQ      .+4                  ;NO
2122 011014 104005                      HLT      !CS1!DA              ;DA MOVED
2123 011016 104414                      CLRDK                      ;CLEAR ALL ERRORS
2124 011020 005777 170074      TST      @RSE                      ;DID ERRORS CLEAR
2125 011024 001401                      BEQ      .+4                  ;YES
2126 011026 104040                      HLT      !DS                  ;ILF DID NOT CLEAR
2127 011030 022777 004200 170046      CMP      #4200,@RSCS1        ;DID ERRORS IN CS1 CLEAR
2128 011036 001401                      BEQ      .+4                  ;YES
2129 011040 104040                      HLT      !DS
2130 011042 000137 010614      JMP      3$                  ;CONTINUE UNTIL DONE
2131 011046                      ILFDN:                          ;DONE WITH ILLEGAL FUNCTION TEST
2132                      ;*****
2133                      ;TEST 61                      ILLEGAL FUNCTION CODE TEST CODE 53 TO 77
2134                      ;*****
2135 011046 104400      TST61: SCOPE
2136
2137                      ;TEST ILF BIT IN RSE AND ERR BIT IN RSDS
2138                      ;ALSO CHECKS TO SEE IF WC,BA, OR DA GOT MODIFIED
2139                      ;IF WISHING TO LOOP ON ONE FUNCTION ONLY, LOAD
2140                      ;FUNCTION INTO LOCATION ILFTB2: AND 0 IN FOLLOWING LOCATION
2141
2142 011050 104414      ILLFUN: CLRDK                      ;CLEAR ALL RS REG
2143 011052 012703 027506      1$: MOV      #ILFTB2,R3          ;GET TABLE OF ILL FUNS.
2144 011056 005037 001176      3$: CLR      WORK              ;CLEAR WORK
2145 011062 012300                      MOV      (R3)+,BAD          ;GET ILL FUN
2146 011064 001554                      BEQ      ILFDNE              ;DONE GET OUT
2147 011066 013777 001102 170016      MOV      @#OBUFSV,@RSBA      ;SET UP REGS.
2148 011074 012777 177777 170006      MOV      #-1,@RSWC          ;TO CHECK FOR CHANGE
2149 011102 010037 001200                      MOV      BAD,WORK1          ;SHOULD WE TEST
2150 011106 042737 177707 001200      BIC      #177707,WORK1        ;BA AND WC
2151 011114 022737 000060 001200      CMP      #60,WORK1          ;TO INC
2152 011122 001003                      BNE      2$                  ;NO
2153 011124 012737 000007 001176      MOV      #7,WORK            ;YES
2154 011132 010077 167746      2$: MOV      BAD,@RSCS1        ;DO ILLEGAL FUNCTION
2155 011136 042700 000001      10$: BIC      #BIT0,BAD          ;CLEAR GO BIT
2156 011142 010001                      MOV      BAD,GOOD          ;MOV ILLEGAL FUN INTO GOOD
```



```
2157 011144 105777 167734      6$: TSTB @RSCS1 ;RDY SET?
2158 011150 100375      BPL 6$ ;NO
2159 011152 052701 144200      BIS #144200,GOOD ;SET ERROR BITS
2160 011156 017700 167722      MOV @RSCS1,BAD ;PUT CS1 INTO BAD
2161 011162 020100      CMP GOOD,BAD ;IS CS1 CORRECT?
2162 011164 001401      BEQ .+4 ;YES
2163 011166 104000      HLT ;GOOD IS WHAT CS1 SHOULD =BAD=CS1
2164 011170 022777 000001 167722      CMP #1,@RSER ;DID ILF SET?
2165 011176 001401      BEQ .+4 ;YES
2166 011200 104043      HLT !CS1!ER!DS ;ILF DID NOT SET
2167 011202 022777 150600 167706      CMP #150600,@RSDS ;IS DS GOOD?
2168 011210 001401      BEQ .+4 ;YES
2169 011212 104043      HLT !CS1!ER!DS ;ERR DID NOT SET
2170 011214 005777 167674      TST @RSDA ;DID DA MOVE?
2171 011220 001401      BEQ .+4 ;NO
2172 011222 104005      HLT !CS1!DA ;DA MOVED
2173 011224 005737 001176      TST WORK ;IS THIS AN ILL WRITE FUN?
2174 011230 001025      BNE 11$ ;YES
2175 011232 017700 167650      MOV @RSCS2,BAD ;GET CS2
2176 011236 013701 001152      MOV UNNUM,GOOD ;GET UNIT #
2177 011242 052701 001100      BIS #1100,GOOD ;SET IR BIT
2178 011246 020100      CMP GOOD,BAD ;IS CS2 CORRECT?
2179 011250 001401      BEQ .+4 ;YES
2180 011252 104000      HLT ;GOOD = CORRECT ANS FOR CS2
2181 011254 023777 001102 167630      CMP @#OBUFSV,@RSBA ;DID BA MOVE
2182 011262 001401      BEQ .+4 ;NO
2183 011264 104021      HLT !CS1!BA ;BA MOVED ON AN ILLEGAL FUNCTION
2184 011266 022777 177777 167614      CMP #-1,@RSWC ;DID WC MOVE?
2185 011274 001401      BEQ .+4 ;NO
2186 011276 104011      HLT !CS1!WC ;WC MOVED
2187 011300 000137 011366      JMP 4$ ;CONTINUE UNTIL DONE
2188 011304 017700 167576      11$: MOV @RSCS2,BAD ;GET CS2
2189 011310 013701 001152      MOV UNNUM,GOOD ;GET UNIT #
2190 011314 052701 001300      BIS #1300,GOOD ;SET IR BIT
2191 011320 020100      CMP GOOD,BAD ;IS CS2 CORRECT?
2192 011322 001401      BEQ .+4 ;YES
2193 011324 104000      HLT ;GOOD = CORRECT ANS FOR CS2
2194 011326 013737 001102 001176      MOV @#OBUFSV,WORK ;GET BUFFER ADDR.
2195 011334 062737 000002 001176      ADD #2,WORK ;UPDATE IT
2196 011342 023777 001176 167542      CMP WORK,@RSBA ;DID BA MOVE
2197 011350 001401      BEQ .+4 ;YES
2198 011352 104021      HLT !CS1!BA ;BA MOVED ON AN ILLEGAL FUNCTION
2199 011354 022777 000000 167526      CMP #0,@RSWC ;DID WC MOVE?
2200 011362 001401      BEQ .+4 ;NO
2201 011364 104011      HLT !CS1!WC ;WC MOVED
2202 011366 104414      4$: CLRDK ;CLEAR ALL ERRORS
2203 011370 022777 004200 167506      CMP #4200,@RSCS1 ;DID ERRORS CLEAR
2204 011376 001401      BEQ .+4 ;YES
2205 011400 104040      HLT !DS ;NO
2206 011402 005777 167512      TST @RSER ;DID ERROR CLEAR
2207 011406 001401      BEQ .+4 ;NO
2208 011410 104040      HLT !DS ;YES
2209 011412 000137 011056      JMP 3$ ;CONTINUE UNTIL DONE
2210 011416      ILFDNE: ;DONE WITH ILLEGAL FUNCTION TEST
```



```

2211 *****
2212 :TEST 62 TEST ILLEGAL FUNCTION CODE 67
2213 *****
2214 011416 104400 TST62: SCOPE
2215
2216 011420 104414 ILF67: CLRDK ;CLEAR ALL RS REG
2217 011422 012777 177777 167460 MOV #-1,@RSWC ;SET WC TO -1
2218 011430 013737 001102 001176 MOV @#OBUFSV,WORK ;GET OUTBUF ADD.
2219 011436 062737 000002 001176 ADD #2,WORK ;FOR TEST
2220 011444 013777 001176 167440 MOV WORK,@RSBA ;LOAD ADDR.
2221 011452 012777 000067 167424 MOV #67,@RSCS1 ;DO FUNCTION 67
2222 011460 105777 167420 1$: TSTB @RSCS1 ;DONE YET?
2223 011464 100375 BPL 1$ ;NO
2224 011466 017700 167420 MOV @RSBA,BAD ;GET BA REG
2225 011472 013701 001102 MOV @#OBUFSV,GOOD ;GET CORRECT ANS FOR RSBA
2226 011476 020100 CMP GOOD,BAD ;IS RSBA CORRECT?
2227 011500 001401 BEQ .+4 ;YES
2228 011502 104000 HLT ;BAD=RSBA GOOD=CORRECT ANS.
2229 011504 013701 001152 MOV UNNUM,GOOD ;GET UNIT NUMBER
2230 011510 052701 001300 BIS #1300,GOOD ;SET IR AND OR BITS
2231 011514 017700 167366 MOV @RSCS2,BAD ;GET CS2
2232 011520 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2233 011522 001401 BEQ .+4 ;YES
2234 011524 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS
2235 011526 022777 000001 167364 CMP #1,@RSER ;IS RSER CORRECT?
2236 011534 001401 BEQ .+4 ;YES
2237 011536 104002 HLT !ER ;ER IS WRONG
2238 011540 104414 CLRDK ;CLEAR ALL RS REG
2239 011542 005777 167352 TST @RSER ;DID ERROR CLEAR
2240 011546 001401 BEQ .+4 ;YES
2241 011550 104040 HLT !DS ;NO
2242 011552 012777 177700 167330 MOV #-100,@RSWC ;SET WC TO -100
2243 011560 013737 001102 001176 MOV @#OBUFSV,WORK ;GET OUTBUF ADD.
2244 011566 062737 000200 001176 ADD #200,WORK ;FOR TEST
2245 011574 013777 001176 167310 MOV WORK,@RSBA ;LOAD ADDR
2246 011602 012777 000067 167274 MOV #67,@RSCS1 ;DO FUNCTION 67
2247 011610 105777 167270 2$: TSTB @RSCS1 ;DONE YET?
2248 011614 100375 BPL 2$ ;NO

```



```
2249 011616 013701 001102 3$: MOV @#OBUFSV,GOOD ;GET CORRECT ANS.
2250 011622 017700 167264 MOV @RSBA,BAD ;GET BA REG
2251 011626 020100 CMP GOOD,BAD ;IS RSBA CORRECT?
2252 011630 001401 BEQ .+4 ;YES
2253 011632 104000 HLT ;BAD=RSBA GOOD=CORRECT ANS.
2254 011634 013701 001152 MOV UNNUM,GOOD ;GET UNIT NUMBER
2255 011640 052701 001300 BIS #1300,GOOD ;SET IR AND OR BITS
2256 011644 017700 167236 MOV @RSCS2,BAD ;GET CS2
2257 011650 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2258 011652 001401 BEQ .+4 ;YES
2259 011654 104000 HLT ;BAD=CS2 GOOD=CORRECT ANS
2260 011656 022777 000001 167234 CMP #1,@RSER ;IS RSER CORRECT?
2261 011664 001401 BEQ .+4 ;YES
2262 011666 104002 HLT !ER ;ER IS WRONG
2263 011670 012777 040011 167206 MOV #40011,@RSCS1 ;CLEAR ERRORS
2264 011676 022777 004210 167200 CMP #4210,@RSCS1 ;DID THEY CLEAR IN CS1
2265 011704 001401 BEQ .+4 ;YES
2266 011706 104040 HLT !DS ;NO
2267 011710 005777 167204 TST @RSER ;DID RSER CLEAR
2268 011714 001401 BEQ .+4 ;YES
2269 011716 104040 HLT !DS ;NO
2270 011720 005777 167176 TST @RSAS ;DID RSAS CLEAR
2271 011724 001401 BEQ .+4 ;YES
2272 011726 104100 HLT !AS ;NO
```

```
2273
2274 *****
2275 :TEST 63 TEST PAR IN RSER
2276 *****
```

```
2277 011730 104400 TST63: SCOPE
2278
2279 011732 013737 001164 022560 PARTST: MOV TIMSV,TIMES ;RESTORE LOOP #
2280 011740 104414 CLRDK ;CLEAR ALL RS REG
2281 011742 012777 000010 167150 1$: MOV #10,@RSER ;SET PAR
2282 011750 022777 150600 167140 CMP #150600,@RSDS ;DID ERR,ATA AND DRY SET?
2283 011756 001401 BEQ .+4 ;YES
2284 011760 104042 HLT !DS!ER ;ER SHOULD SET IF PAR SETS IN RSER
2285 011762 104414 CLRDK ;CLEAR ALL RS REG
2286 011764 005777 167130 TST @RSER ;DID PAR CLEAR?
2287 011770 001401 BEQ .+4 ;YES
2288 011772 104002 HLT !ER ;PAR DID NOT CLEAR BY CLEAR BIT
2289 011774 022777 010600 167114 CMP #10600,@RSDS ;DID ERROR BITS CLEAR
2290 012002 001401 BEQ .+4 ;YES
2291 012004 104040 HLT !DS ;NO
```


CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 N 4
TST63 TEST PAR IN RSER PAGE 53

SEQ 0052

```
2292 ;CHECK BITS 12 TO 15 FOR 0
2293 ;CHECK SECTOR FRACTION TO WATCH FOR MOVEMENT
2294 ;CHECK CS BITS IN LA AND ADDRESS CONFIRM IN MR REG
2295
2296 ;*****
2297 ;TEST 64 LOOK AHEAD TEST
2298 ;*****
2299 012006 104400 TST64: SCOPE
2300
2301 012010 032777 170000 167106 LATST: BIT #170000,@RSLA ;ARE BITS 12 TO 15 CLEARED?
2302 012016 001401 BEQ +4 ;YES
2303 012020 104204 HLT !LA ;BITS 12 TO 15 SHOULD BE CLEARED
2304 012022 104400 TST65: SCOPE
2305
2306 ;NOW TEST MOVEMENT IN SF BITS
2307
2308 012024 012737 171005 001176 MOV #171005,WORK ;SET UP FOR TIME OUT ROUTINE
2309 012032 017701 167066 MOV @RSLA,GOOD ;GET READING FROM LA
2310 012036 042701 007700 BIC #7700,GOOD ;GET RID OF CS BITS
2311 012042 005337 001176 1$: DEC WORK ;WAIT FOR DISK
2312 012046 001407 BEQ ERRR ;TYPE ERROR
2313 012050 017700 167050 MOV @RSLA,BAD ;READ LA
2314 012054 042700 007700 BIC #7700,BAD ;CLEAR CS BITS
2315 012060 020100 CMP GOOD,BAD ;DID SF BITS CHANGE?
2316 012062 001767 BEQ 1$ ;WAIT FOR TIME OUT
2317 012064 000422 BR LATDON ;LA OK CONT
2318 012066 ERRR:
2319 012066 104402 012072 TYPE +2 ;.ASCIZ <15><12>'SECTOR FRACTIONS NOT MOVING'
2320 012130 104204 HLT !LA ;TYPE LOOK AHEAD REG
2321 012132 LATDON: ;DONE CONT.
```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 54
TST66 CHECK CS BITS TO INCREMENT AND ADDRESS CONFIRM BIT IN MR

SEQ 0053

```
2322 ;*****
2323 ;TEST 66 CHECK CS BITS TO INCREMENT AND ADDRESS CONFIRM BIT IN MR
2324 ;*****
2325 012132 104400 TST66: SCOPE
2326
2327 012134 013737 022560 001164 CSTST: MOV TIMES,TIMSV ;SAVE LOOP CT
2328 012142 012737 000010 022560 MOV #10,TIMES ;LOOP 10 TIMES
2329 012150 104414 CLRDK ;CLEAR ALL RS REG.
2330 012152 012701 001000 MOV #1000,GOOD ;LOAD COUNTER
2331 012156 032777 001000 166744 BIT #BIT9,@RSMR ;IS ADD CONFIRM BIT 0?
2332 012164 001407 BEQ ADDCF ;YES CONTINUE
2333 012166 005301 DEC GOOD ;WAIT FOR
2334 012170 001376 BNE -2 ;DISK TO MOVE
2335 012172 032777 001000 166730 BIT #BIT9,@RSMR ;IS AADD. CON. BIT BIT 0?
2336 012200 001401 BEQ +4 ;YES
2337 012202 104220 HLT !MR ;ADD. CONF. BIT ALWAYS A 1
2338
2339 ;NOW TEST TA BITS AND ADD. CON. BIT IN MR
2340
2341 012204 012777 177777 166702 ADDCF: MOV #-1,@RSDA ;INIT RSDA
2342 012212 012737 177777 001176 1$: MOV #-1,WORK ;SETUP TIMEOUT COUNTER
2343 012220 005277 166670 INC @RSDA ;GET NEXT SECTOR
2344 012224 022777 010000 166662 CMP #10000,@RSDA ;DONE ALL YET?
2345 012232 001433 BEQ DONCS ;YES
2346 012234 005337 001176 2$: DEC WORK ;DO TIMEOUT ROUTINE
2347 012240 001427 BEQ TIMEOUT ;ADD. CON. NEVER CAME UP
2348 012242 032777 001000 166660 BIT #1000,@RSMR ;DID ADD CONFIRM BIT SET?
2349 012250 001771 BEQ 2$ ;YES
2350 012252 017700 166646 MOV @RSLA,BAD ;GET LA
2351 012256 042700 000077 BIC #77,BAD ;CLEAR SF BITS
2352 012262 012737 000006 001200 MOV #6,WORK1 ;SET UP COUNTER
2353 012270 006000 3$: ROR BAD ;MOV SA BITS RIGHT
2354 012272 005337 001200 DEC WORK1 ;DO 6 TIMES
2355 012276 001374 BNE 3$ ;DONE YET?
2356 012300 017701 166610 MOV @RSDA,GOOD ;GET DA
2357 012304 042701 177700 BIC #177700,GOOD ;CLEAR JUNK
2358 012310 020100 CMP GOOD,BAD ;ARE SA BITS = IN DA AND LA REG.?
2359 012312 001401 BEQ +4 ;OK
2360 012314 104000 HLT ;GOOD =DA BAD = LA
2361 012316 000735 BR 1$ ;NO WAIT
2362
2363 012320 104262 TIMEOUT: HLT !MR!ER!DS ;ADDRESS CONFIRM BIT NEVER SET COULD BE BAD OR
2364 2364 ;BAD LA OR BAD COMPARE BETWEEN LA AND DA
2365 012322 DONCS: ;TEST DONE CONTINUE
```



```
2366 ;*****
2367 ;TEST 67 PARITY TEST
2368 ;*****
2369 012322 104400 TST67: SCOPE
2370
2371 012324 012701 031130 PART: MOV #PARITY , R1 ;GET TEST LOCATION
2372 012330 012737 012354 000004 MOV #1$,a#4 ;SETUP FOR TIMEOUT
2373 012336 012777 000014 166630 MOV #14,@CACHE ;DISABLE CACHE
2374 012344 012737 000001 031132 MOV #1,CACH ;CACHE IS AVAILABLE
2375 012352 000401 BR 2$ ;GET OVER STACK POP
2376 012354 022626 1$: CMP (SP)+,(SP)+ ;NO CACHE, RESET STACK
2377 012356 013737 001164 022560 2$: MOV TIMSV,TIMES ;RESTORE LOOP COUNTER
2378 012364 012737 012566 000004 MOV #PRTP,a#4 ;SETUP TIME OUT VECTOR
2379 012372 012737 000340 000006 MOV #340,a#6 ;SETUP PRIORITY OF 7
2380 012400 012702 172100 MOV #MPRO,R2 ;GET PAR REG ADDR
2381 012404 005712 TSTAGN: TST (R2) ;DOES PAR REG EXIST?
2382 012406 012712 000004 MOV #WWP,a#2 ;YES SET WRITE WRONG PARITY
2383 012412 012737 012426 000004 MOV #6$,a#4 ;SETUP FOR TRAP IF NO MEM MGMT
2384 012420 005037 177572 CLR SRO ;TURN OFF MEM MGMT(IF THERE)
2385 012424 000401 BR 1$ ;GET AROUND STACK POP(MEM MGMT THERE)
2386 012426 022626 6$: CMP (SP)+,(SP)+ ;NO MEM MGMT, POP STACK
2387 012430 011111 1$: MOV @R1,@R1 ;WRITE WRONG PARITY
2388 012432 005711 TST @R1 ;READ IT
2389 012434 005712 TST @R2 ;DID PARITY ERROR SET?
2390 012436 100402 BMI 2$ ;IF YES, THEN BRANCH
2391 012440 005012 CLR @R2 ;CLEAR PARITY REG
2392 012442 000452 BR PRTP1 ;GET NEXT PARITY REG
2393 012444 042712 100004 2$: BIC #100004,@R2 ;TURN OFF WWP,PARITY ERROR
2394 012450 052712 000001 BIS #1,@R2 ;SET PARITY ERROR IND. ENABLE
2395 012454 012737 012564 000114 MOV #5$,a#114 ;SET UP FOR PARITY TRAP
2396 012462 104414 CLRDK ;CLEAR ALL RS REG
2397 012464 012777 031130 166420 MOV #PARITY,@RSBA ;SET UP CURRENT ADDRESS
2398 012472 012777 177777 166410 MOV #-1,@RSWC ;SET WORD COUNT TO -1
2399 012500 012777 000061 166376 MOV #61,@RSCS1 ;GO WRITE
2400 012506 105777 166372 3$: TSTB @RSCS1 ;DONE YET?
2401 012512 100375 BPL 3$ ;NO WAIT
2402 012514 017700 MOV @RSCS2,BAD ;GET CS2
2403 012520 012701 020100 MOV #20100,GOOD ;EXPECTED RESULTS
2404 012524 053701 001152 BIS UNNUM,GOOD ;GET CORRECT UNIT # FOR CS2
2405 012530 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2406 012532 001401 BEQ .+4 ;YES
2407 012534 104000 HLT ;CS2 SHOULD = GOOD
2408 012536 022777 144260 166340 CMP #144260,@RSCS1 ;IS CS1 CORRECT?
2409 012544 001401 BEQ .+4 ;YES
2410 012546 104040 HLT ;CS1 SHOULD = 144260
2411 012550 005737 031132 TST CACH ;DO WE HAVE CACHE?
2412 012554 001402 BEQ 4$ ;BRANCH IF NOT
2413 012556 005077 166412 CLR @CACHE ;ENABLE CACHE
2414 012562 000411 4$: BR NOPAR ;GET OUT
2415 012564 000002 5$: RTI ;FOR PARITY ERRORS
2416
2417 ;TRAPOUT ROUTINE
2418
2419 012566 022626 PRTP: CMP (6)+,(6)+ ;CLEAR STACK
2420 012570 022702 172136 PRTP1: CMP #172136,R2 ;DONE YET?
2421 012574 001404 BEQ NOPAR ;YES NO PAR REG
```


CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 56
TST67 PARITY TEST

SEQ 0055

```
2422 012576 062702 000002          ADD    #2,R2          ;NO TRY AGAIN
2423 012602 000137 012404          JMP     TSTAGN        ;RETRY
2424
2425 012606 012737 000006 000004 NOPAR: MOV    #6,#4
2426 012614 005037 000006          CLR     @#6
2427 012620 005037 031130          CLR     PARITY      ;WRITE GOOD PARITY
2428          ;*****
2429          ;TEST 70          TEST WRITE CHECK ERROR
2430          ;*****
2431 012624 104400          TST70: SCOPE
2432
2433          ;WRITE A WORD OF 0 AND FLOAT A 1 TRSOUGH IT TO CAUSE WCE
2434          ;SET BIT14 IN ONCEE AND WRITE A WD OF -1 AND FLOAT 0
2435          ;TO CAUSE WCE
2436
2437 012626 104414          WCETST: CLRDK          ;CLEAR ALL RS REG
2438 012630 005037 027530          CLR     OUTBUF        ;WRITE A WD OF 0
2439 012634 013777 001102 166250 WCETT: MOV    @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
2440 012642 012777 177777 166240          MOV    #-1,@RSWC    ;SET WORD COUNT TO -1
2441 012650 012777 000061 166226          MOV    #61,@RSCS1   ;GO WRITE
2442 012656 105777 166222          3$:    TSTB    @RSCS1   ;DONE YET?
2443 012662 100375          BPL     3$          ;NO WAIT
2444 012664 032737 040000 001160          BIT     #BIT14,ONCEE ;WRITE A 1 OR 0?
2445 012672 001410          BEQ     2$          ;WRITE A 0
2446 012674 012737 177777 027530          MOV    #-1,OUTBUF    ;WRITE A 1
2447 012702 000241          CLC          ;CLEAR CARRY
2448 012704 006137 027530          6$:    ROL     OUTBUF    ;FLOAT A 0 THROUGH BAD WD
2449 012710 103123          BCC     WCEDON        ;DONE GET OUT
2450 012712 000406          BR      5$          ;CHECK WCE
2451 012714 005037 027530          2$:    CLR     OUTBUF    ;WRITE A 0
2452 012720 000261          SEC          ;SET CARRY
2453 012722 006137 027530          1$:    ROL     OUTBUF    ;FLOAT A 1
2454 012726 103503          BCS     WCEDNE        ;GET OUT WHEN DONE
2455 012730 013777 001102 166154          5$:    MOV    @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
2456 012736 012777 177777 166144          MOV    #-1,@RSWC    ;SET WORD COUNT TO -1
2457 012744 005077 166144          CLR     @RSDA
2458 012750 012777 000051 166126          MOV    #51,@RSCS1   ;GO WRITE CHECK
2459 012756 105777 166122          4$:    TSTB    @RSCS1   ;READY YET?
2460 012762 100375          BPL     4$          ;NO WAIT
2461 012764 017700 166116          MOV    @RSCS2,BAD    ;GET CS2
2462 012770 013701 001152          MOV    UNNUM,GOOD    ;SET UNIT #
2463 012774 052701 040300          BIS     #40300,GOOD  ;SET BITS
2464 013000 020100          CMP     GOOD,BAD    ;IS CS2 CORRECT?
2465 013002 001413          BEQ     7$          ;YES
2466 013004 104000          HLT          ;BAD=CS2 GOOD=CORRECT ANS
2467 013006 013700 027530          MOV    OUTBUF,BAD    ;GET BAD WD THAT SHOULD CAUSE WCE
2468 013012 005001          CLR     GOOD        ;GET GOOD WD IF WRITING 0
2469 013014 032737 040000 001160          BIT     #BIT14,ONCEE ;ARE WE WRITING 1 OR 0
2470 013022 001402          BEQ     8$          ;0
2471 013024 012701 177777          MOV    #-1,GOOD    ;GET GOOD WD FOR 1
2472 013030 104000          8$:    HLT          ;GOOD = CORRECT WD WRITTEN
2473
2474 013032 022777 144250 166044          7$:    CMP     #144250,@RSCS1 ;BAD = INCORRECT WD THAT WCE DID NOT CATCH
2475 013040 001401          BEQ     +4          ;DID TRE SET?
2476 013042 104043          HLT          ;YES
2477 013044 017700 166042          MOV    @RSBA,BAD    ;TRE SHOULD SET IF WCE SETS
                        ;FETCH CURRENT ADDRESS
```


2478	013050	013701	001102		MOV	@#OBUFSV,GOOD	;WHAT RSBA SHOULD EQUAL
2479	013054	062701	000002		ADD	#2,GOOD	;UPDATE IT
2480	013060	020001			CMP	BAD,GOOD	;IS RSBA CORRECT
2481	013062	001401			BEQ	+.4	;YES EXECUTE CONTINUE
2482	013064	104000			HLT		;RSBA FAILED TO INCREMENT
2483	013066	104414			CLRDK		;CLEAR ALL RS REG
2484	013070	013701	001152		MOV	UNNUM,GOOD	;PUT DRIVE IN GOOD
2485	013074	052701	000100		BIS	#100,GOOD	;SET IR BIT
2486	013100	017700	166002		MOV	@RSCS2,BAD	;GET CS2
2487	013104	020100			CMP	GOOD,BAD	;IS CS2 CORRECT
2488	013106	001401			BEQ	+.4	;YES
2489	013110	104000			HLT		;BAD =CS2 GOOD IS CORRECT ANS
2490	013112	022777	004200	165764	CMP	#4200,@RSCS1	;DID TRE CLEAR?
2491	013120	001401			BEQ	+.4	;YES
2492	013122	104001			HLT	!CS1	;TRE DID NOT CLEAR WITH CLEAR
2493	013124	032737	040000	001160	BIT	#BIT14,ONCEE	;FLOATION A 1 OR 0?
2494	013132	001673			BEQ	1\$	;FLOAT 1
2495	013134	000663			BR	6\$	;FLOAT 0
2496	013136	052737	040000	001160	WCEDNE: BIS	#BIT14,ONCEE	;SET BIT14
2497	013144	104414			CLRDK		
2498	013146	012737	177777	027530	MOV	#-1,OUTBUFF	
2499	013154	000137	012634		JMP	WCETT	;NOW WRITE -1 IN OUTBUF
2500	013160	042737	040000	001160	WCEDON: BIC	#BIT14,ONCEE	;CLEAR TEST FLAG


```

2501 ;*****
2502 ;TEST 71 TEST WRITE CHECK ERROR ON -B- PORT
2503 ;*****
2504 013166 104400 TST71: SCOPE
2505
2506 ;WRITE A WORD OF 0 AND FLOAT A 1 TRSOUGH IT TO CAUSE WCE
2507 ;SET BIT14 IN ONCEE AND WRITE A WD OF -1 AND FLOAT 0
2508 ;TO CAUSE WCE
2509
2510 013170 032737 020000 001160 WCETSB: BIT #BIT13,ONCEE ;:-B- PORT?
2511 013176 001402 BEQ 1$ ;YES
2512 013200 000137 013544 JMP WCEDOS ;NO GET OUT
2513 013204 013737 001170 001102 1$: MOV BPORTT,OBUSV ;GET -B- PORT BUFFER
2514 013212 104414 CLRDK ;CLEAR ALL RS REG
2515 013214 005077 165750 CLR @BPORTT ;WRITE A WD OF 0
2516 013220 013777 001102 165664 WCETB: MOV @#OBUSV,@RSBA ;SET UP CURRENT ADDRESS
2517 013226 012777 177777 165654 MOV #-1,@RSWC ;SET WORD COUNT TO -1
2518 013234 012777 002061 165642 MOV #2061,@RSCS1 ;GO WRITE
2519 013242 105777 165636 3$: TSTB @RSCS1 ;DONE YET?
2520 013246 100375 BPL 3$ ;NO WAIT
2521 013250 032737 040000 001160 BIT #BIT14,ONCEE ;WRITE A 1 OR 0?
2522 013256 001410 BEQ 2$ ;WRITE A 0
2523 013260 012777 177777 165702 MOV #-1,@BPORTT ;WRITE A 1
2524 013266 000241 CLC ;CLEAR CARRY
2525 013270 006137 027530 6$: ROL OUTBUF ;FLOAT A 0 THROUGH BAD WD
2526 013274 103123 BCC WCEDOS ;DONE GET OUT
2527 013276 000406 BR 5$ ;CHECK WCE
2528 013300 005077 165664 2$: CLR @BPORTT ;WRITE A 0
2529 013304 000261 SEC ;SET CARRY
2530 013306 006177 165656 1$: ROL @BPORTT ;FLOAT A 1
2531 013312 103503 BCS WCEDNB ;GET OUT WHEN DONE
2532 013314 013777 001102 165570 5$: MOV @#OBUSV,@RSBA ;SET UP CURRENT ADDRESS
2533 013322 012777 177777 165560 MOV #-1,@RSWC ;SET WORD COUNT TO -1
2534 013330 005077 165560 CLR @RSDA
2535 013334 012777 002051 165542 MOV #2051,@RSCS1 ;GO WRITE CHECK
2536 013342 105777 165536 4$: TSTB @RSCS1 ;READY YET?
2537 013346 100375 BPL 4$ ;NO WAIT
2538 013350 017700 165532 MOV @RSCS2,BAD ;GET CS2
2539 013354 013701 001152 MOV UNNUM,GOOD ;SET UNIT #
2540 013360 052701 040300 BIS #40300,GOOD ;SET BITS
2541 013364 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
2542 013366 001413 BEQ 7$ ;YES
2543 013370 104000 HLT ;CS2=BAD GOOD=CORRECT ANS
2544 013372 017700 165572 MOV @BPORTT,BAD ;GET BAD WD THAT SHOULD CAUSE WCE
2545 013376 005001 CLR GOOD ;GET GOOD WD IF WRITING 0
2546 013400 032737 040000 001160 BIT #BIT14,ONCEE ;ARE WE WRITING 1 OR 0
2547 013406 001402 BEQ 8$ ;0
2548 013410 012701 177777 MOV #-1,GOOD ;GET GOOD WD FOR 1
2549 013414 104000 8$: HLT ;GOOD = CORRECT WD WRITTEN
2550 ;BAD = INCORRECT WD THAT WCE DID NOT CATCH

```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 59
TST71 TEST WRITE CHECK ERROR ON -B- PORT

SEQ C058

```
2551 013416 022777 146250 165460 7$: CMP #146250,@RSCS1 ;DID TRE SET?
2552 013424 001401 BEQ .+4 ;YES
2553 013426 104043 HLT !CS1!ER!DS ;TRE SHOULD SET IF WCE SETS
2554 013430 017700 165456 MOV @RSBA,BAD ;FETCH CURRENT ADDRESS
2555 013434 013701 001102 MOV @#OBUFSV,GOOD ;WHAT RSBA SHOULD EQUAL
2556 013440 062701 000002 ADD #2,GOOD ;UPDATE IT
2557 013444 020001 CMP BAD,GOOD ;IS RSBA CORRECT
2558 013446 001401 BEQ .+4 ;YES EXECUTE CONTINUE
2559 013450 104000 HLT ;RSBA FAILED TO INCREMENT
2560 013452 104414 CLRDK ;CLEAR ALL RS REG
2561 013454 013701 001152 MOV UNNUM,GOOD ;PUT DRIVE IN GOOD
2562 013460 052701 000100 BIS #100,GOOD ;SET IR BIT
2563 013464 017700 165416 MOV @RSCS2,BAD ;GET CS2
2564 013470 020100 CMP GOOD,BAD ;IS CS2 CORRECT
2565 013472 001401 BEQ .+4 ;YES
2566 013474 104000 HLT ;BAD =CS2 GOOD IS CORRECT ANS
2567 013476 022777 004200 165400 CMP #4200,@RSCS1 ;DID TRE CLEAR?
2568 013504 001401 BEQ .+4 ;YES
2569 013506 104001 HLT !CS1 ;TRE DID NOT CLEAR WITH CLEAR
2570 013510 032737 040000 001160 BIT #BIT14,ONCEE ;FLOATION A 1 OR 0?
2571 013516 001673 BEQ 1$ ;FLOAT 1
2572 013520 000663 BR 6$ ;FLOAT 0
2573 013522 052737 040000 001160 WCEDNB: BIS #BIT14,ONCEE ;SET BIT14
2574 013530 104414 CLRDK
2575 013532 012777 177777 165430 MOV #-1,@BPORTT
2576 013540 000137 013220 JMP WCETB ;NOW WRITE -1 IN OUTBUF
2577 013544 012737 027530 001102 WCEDOS: MOV #OUTBUF,OBUFSV ;RESTORE OBUFSV
2578 013552 042737 040000 001160 BIC #BIT14,ONCEE ;CLEAR TEST FLAG
```



```

2579
2580
2581
2582 013560 104400
2583
2584 013562 104414
2585 013564 012737 177777 027530
2586 013572 013777 001102 165312
2587 013600 012777 177000 165302
2588 013606 012777 000061 165270
2589 013614 105777 165264
2590 013620 100775
2591 013622 012777 000071 165254
2592 013630 004737 026744
2593 013634 104001
2594 013636 022777 144260 165240
2595 013644 001401
2596 013646 104001
2597 013650 005777 165246
2598 013654 001401
2599 013656 104100
2600 013660 013701 001152
2601 013664 052701 002300
2602 013670 017700 165212
2603 013674 020100
2604 013676 001401
2605 013700 104000
2606 013702 005777 165202
2607 013706 001001
2608 013710 104011
2609 013712 005777 165202
2610 013716 001401
2611 013720 104040
2612 013722 052777 040000 165154
2613 013730 042701 002000
2614 013734 017700 165146
2615 013740 020100
2616 013742 001401
2617 013744 104200
2618 013746 022777 004260 165130
2619 013754 001401
2620 013756 104040

;*****
;TEST 72 TEST PROGRAM ERROR BIT IN RSCS2
;*****
TST72: SCOPE

PGETST: CLRDK
MOV #177777,OUTBUF ;CLEAR ALL RS REG
MOV @#OBUFSV,@RSBA ;DATA TO BE X-FERED
MOV #177000,@RSWC ;SET UP CURRENT ADDRESS
MOV #61,@RSCS1 ;SET WORD COUNT
2$: TSTB @RSCS1 ;GO WRITE
BMI 2$ ;IS RDY CLEARED YET?
MOV #71,@RSCS1 ;NO WAIT
JSR PC,WAITRY ;GO READ
HLT !CS1 ;WAIT FOR READY
CMP #144260,@RSCS1 ;RDY NEVER CAME UP
BEQ .+4 ;IS CS1 CORRECT?
HLT !CS1 ;YES
TST @RSAS ;TRE SHOULD SET BY SETTING PGE
BEQ .+4 ;AS SHOULD = 0
HLT !AS ;YES
MOV UNNUM,GOOD ;RSAS SHOULD = 0
BIS #2300,GOOD ;GET UNIT #
MOV @RSCS2,BAD ;SET PGE, IR, AND OR
CMP GOOD,BAD ;GET CS2
BEQ .+4 ;IS IT CORRECT?
HLT !AS ;YES
TST @RSWC ;BAD = CS2
BNE .+4 ;SHOULD NOT BE 0
HLT !WC!CS1 ;BECAUSE PGE SHOULD ABORT
TST @RSER ;CURRENT OPERATION
BEQ .+4 ;DID ANY ERRORS SET?
HLT !DS ;NO
BIS #TRE,@RSCS1 ;RMR SHOULD BE SET
BIC #PGE,GOOD ;CLEAR ERRORS
MOV @RSCS2,BAD ;CLEAR PGE ERROR
CMP GOOD,BAD ;GET CS2
BEQ .+4 ;IS CS2 CORRECT?
HLT !CS2 ;YES
CMP #4260,@RSCS1 ;PGE DID NOT CLEAR BY CLEARING TRE BAD = CS2
BEQ .+4 ;DID SC CLEAR
HLT !DS ;YES
;DID NOT CLEAR BY CLEARING TRE

```



```
2621
2622
2623
2624 013760 104400
2625
2626 013762 104414
2627 013764 013777 001102 165120
2628 013772 012777 177700 165110
2629 014000 012703 172060
2630 014004 011304
2631 014006 042704 000017
2632 014012 022704 000020
2633 014016 001372
2634 014020 012777 000031 165056
2635 014026 012777 007777 165060
2636 014034 004737 026744
2637 014040 104001
2638 014042 022777 000004 165050
2639 014050 001401
2640 014052 104002
2641 014054 022777 007777 165032
2642 014062 001001
2643 014064 104004
2644 014066 022777 150600 165022
2645 014074 001401
2646 014076 104042
2647 014100 022777 104230 164776
2648 014106 001401
2649 014110 104040
2650 014112 104414
2651 014114 005777 165000
2652 014120 001401
2653 014122 104002
2654 014124 022777 004200 164752
2655 014132 001401
2656 014134 104040
2657
2658
2659
2660
2661 014136 104400
2662
2663 014140 104414
2664 014142 013777 001102 164742
2665 014150 012777 177700 164732
2666 014156 012777 000061 164720
2667 014164 105777 164714
2668 014170 100775
2669 014172 012777 177773 164720
```

```
*****
;TEST 73 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSDA
*****
TST73: SCOPE

RMRT1: CLRDK ;CLEAR ALL RS REG
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
MOV #177700,@RSWC ;SET WORD COUNT
MOV #172060,R3 ;GET RSLA REG
1$: MOV @R3,R4 ;WAIT FOR
BIC #17,R4 ;THE MIDDLE
CMP #20,R4 ;OF SECTOR 0
BNE 1$ ;BEFORE DOING A SEARCH
MOV #31,@RSCS1 ;SEARCH
MOV #7777,@RSDA ;CAUSE ERROR
JSR PC,WAITRY ;WAIT FOR READY
HLT !CS1 ;RDY NEVER CAME UP
CMP #4,@RSER ;DID RMR SET?
BEQ .+4 ;YES
HLT !ER ;ER SHOULD = 4
CMP #7777,@RSDA ;DID DA GET MODIFIED?
BNE .+4 ;NO
HLT !DA
CMP #150600,@RSDS ;DID ERR SET?
BEQ .+4 ;YES
HLT !DS!ER ;ER DID NOT SET BECAUSE OF RMR
CMP #104230,@RSCS1 ;IS CS1 CORRECT?
BEQ .+4 ;YES
HLT !DS ;CS1 SHOULD = 144260
CLRDK ;CLEAR ALL RS REG
TST @RSER ;DID RMR CLEAR?
BEQ .+4 ;YES
HLT !ER ;RMR DID NOT CLEAR WITH A CLEAR
CMP #4200,@RSCS1 ;IS CS1 CORRECT?
BEQ .+4 ;YES
HLT !DS ;NO

*****
;TEST 74 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSER
*****
TST74: SCOPE

RMRT2: CLRDK ;CLEAR ALL RS REG
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
MOV #177700,@RSWC ;SET WORD COUNT
MOV #61,@RSCS1 ;GO WRITE
2$: TSTB @RSCS1 ;IS RDY SET?
BMI 2$ ;YES WAIT FOR IT TO CLEAR
MOV #177773,@RSER ;CAUSE ERROR
```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 62

TST74 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSER

SEQ 0061

```
2670 014200 004737 026744 JSR PC, WAITRY
2671 014204 104001 HLT !CS1 ;RDY NEVER CAME UP
2672 014206 022777 000004 164704 CMP #4, @RSER ;DID RMR SET?
2673 014214 001401 BEQ .+4 ;YES
2674 014216 104002 HLT !ER ;ER SHOULD = 4
2675 014220 022777 150600 164670 CMP #150600, @RSDS ;DID ERR SET?
2676 014226 001401 BEQ .+4 ;YES
2677 014230 104042 HLT !DS!ER ;ERR DID NOT SET BECAUSE OF RMR
2678 014232 022777 144260 164644 CMP #144260, @RSCS1 ;IS CS1 CORRECT?
2679 014240 001401 BEQ .+4 ;YES
2680 014242 104040 HLT !DS ;CS1 SHOULD = 144260
2681 014244 104414 CLRDK ;CLEAR ALL RS REG
2682
```

```
2683 ;*****
2684 ;TEST 75 TEST RMR IN RSER REGISTER TRYING TO WRITE INTO RSCS1
2685 ;*****
```

2686 014246 104400

TST75: SCOPE

```
2687
2688 014250 104414 RMRT3: CLRDK ;CLEAR ALL RS REG
2689 014252 013777 001102 164632 MOV @#0BUFSV, @RSBA ;SET UP CURRENT ADDRESS
2690 014260 012777 177700 164622 MOV #177700, @RSWC ;SET WORD COUNT
2691 014266 012777 000061 164610 MOV #61, @RSCS1 ;GO WRITE
2692 014274 105777 164604 2$: TSTB @RSCS1 ;IS RDY SET?
2693 014300 100775 BMI 2$ ;YES WAIT FOR IT TO CLEAR
2694 014302 012777 000030 164574 MOV #30, @RSCS1 ;CAUSE ERROR
2695 014310 004737 026744 JSR PC, WAITRY ;WAIT FOR READY
2696 014314 104001 HLT !CS1 ;RDY NEVER CAME UP
2697 014316 022777 000004 164574 CMP #4, @RSER ;DID RMR SET?
2698 014324 001401 BEQ .+4 ;YES
2699 014326 104002 HLT !ER ;ER SHOULD = 4
2700 014330 022777 150600 164560 CMP #150600, @RSDS ;DID ERR SET?
2701 014336 001401 BEQ .+4 ;YES
2702 014340 104042 HLT !DS!ER ;ERRID NOT SET BECAUSE OF RMR
2703 014342 022777 144260 164534 CMP #144260, @RSCS1 ;IS CS1 CORRECT?
2704 014350 001401 BEQ .+4 ;YES
2705 014352 104040 HLT !DS ;CS1 SHOULD = 144260
2706 014354 104414 CLRDK ;CLEAR ALL RS REG
```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 63
TST76 TEST THAT RMR DOES NOT SET BY WRITTING INTO RSAS

SEQ 0062

```
2707 ;*****
2708 ;TEST 76 TEST THAT RMR DOES NOT SET BY WRITTING INTO RSAS
2709 ;*****
2710 014356 104400 TST76: SCOPE
2711
2712 014360 104414 RMRT4: CLRDK ;CLEAR ALL RS REG
2713 014362 013777 001102 164522 MOV @#0BUFSV,@RSBA ;SET UP CURRENT ADDRESS
2714 014370 012777 177700 164512 MOV #177700,@RSWC ;SET WORD COUNT
2715 014376 012703 172060 MOV #172060,R3 ;GET RSLA REG
2716 014402 011304 1$: MOV @R3,R4 ;WAIT FOR
2717 014404 042704 000017 BIC #17,R4 ;THE MIDDLE
2718 014410 022704 000020 CMP #20,R4 ;OF SECTOR 0
2719 014414 001372 BNE 1$ ;BEFORE DOING A SEARCH
2720 014416 012777 000031 164460 MOV #31,@RSCS1 ;SEARCH
2721 014424 012777 000000 164470 MOV #0,@RSAS ;TRY TO CAUSE ERROR
2722 014432 005037 001176 CLR WORK ;CLEAR COUNTER
2723 014436 032777 000200 164452 2$: BIT #BIT7,@RSDS ;WAIT FOR DRY
2724 014444 001004 BNE 3$ ;READY CONT
2725 014446 005237 001176 INC WORK ;COUNT
2726 014452 001371 BNE 2$ ;RETRY
2727 014454 104001 HLT !CS1 ;RDY NEVER CAME UP
2728 014456 005777 164436 3$: TST @RSE ;DID RMR SET?
2729 014462 001401 BEQ .+4 ;NO
2730 014464 104002 HLT !ER ;ER SHOULD = 0
2731 014466 022777 110600 164422 CMP #110600,@RSDS ;DID ERR SET?
2732 014474 001401 BEQ .+4 ;NO
2733 014476 104042 HLT !DS!ER ;DS SHOULD = 110600
2734 014500 022777 104230 164376 CMP #104230,@RSCS1 ;IS CS1 CORRECT?
2735 014506 001401 BEQ .+4 ;YES
2736 014510 104040 HLT !DS ;CS1 SHOULD = 144260
2737 014512 104414 CLRDK ;CLEAR ALL RS REG
2738 014514 022777 004200 164362 CMP #4200,@RSCS1 ;IS CS1 CORRECT?
2739 014522 001401 BEQ .+4 ;YES
2740 014524 104040 HLT !DS ;NO
```



```
2741 :*****
2742 :TEST 77 TEST DCK IN RSER
2743 :*****
2744 014526 104400 TST77: SCOPE
2745
2746 ;DO A WRITE AND THEN A CLEAR FUNCTION THAT SHOULD CAUSE DCK TO SET
2747
2748 014530 104414 DCKTST: CLRDK ;CLEAR ALL RS REG
2749 014532 022737 000004 001162 CMP #4,RS04DT ;IS THIS A LA DISK?
2750 014540 001004 BNE 7$ ;NO
2751 014542 012737 177640 001202 MOV #177640,WORK2 ;GET WC FOR LA DISK
2752 014550 000411 BR 1$ ;CONTINUE
2753 014552 012737 177500 001202 7$: MOV #177500,WORK2 ;GET WC FOR RS04
2754 014560 005737 001162 TST RS04DT ;IS THIS A RS04?
2755 014564 001003 BNE 1$ ;YES
2756 014566 012737 177600 001202 MOV #177600,WORK2 ;NO
2757 014574 013777 001202 164306 1$: MOV WORK2,@RSWC ;LOAD WC
2758 014602 012737 177777 027530 MOV #-1,OUTBUF ;WRITE -1
2759 014610 013777 001102 164274 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
2760 014616 052777 000010 164262 4$: BIS #10,@RSCS2 ;SET BAI BIT
2761 014624 012777 000061 164252 MOV #61,@RSCS1 ;GO WRITE
2762 014632 105777 164246 5$: TSTB @RSCS1 ;IS RDY SET?
2763 014636 100375 BPL 5$ ;WAIT FOR WRITE TO FINISH
2764 014640 005077 164250 2$: CLR @RSDA ;SET DSK ADDRESS TO 0
2765 014644 005037 027530 CLR OUTBUF ;WRITE 0
2766 014650 013777 001102 164234 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
2767 014656 012777 177777 164224 MOV #-1,@RSWC ;LOAD WC
2768 014664 012702 172060 MOV #172060,R2 ;PUT RSLA ADDR INTO R2
2769 014670 011203 3$: MOV (R2),R3 ;GET LA AND WAIT FOR
2770 014672 042703 000077 BIC #77,R3 ;SECTOR 40
2771 014676 022703 004000 CMP #4000,R3 ;BEFORE
2772 014702 001372 BNE 3$ ;WRITING
2773 014704 012777 000061 164172 MOV #61,@RSCS1 ;GO WRITE
2774 014712 011203 6$: MOV (R2),R3 ;GET RSLA AND WAIT FOR
2775 014714 042703 000017 BIC #17,R3 ;MIDDLE OF SECTOR
2776 014720 022703 000020 CMP #20,R3 ;0 BEFORE EXECUTING
2777 014724 001372 BNE 6$ ;A CLEAR FUNCTION
2778 014726 012777 000040 164152 MOV #40,@RSCS2 ;CLEAR ALL REG. DO IT THIS WAY
2779 014734 013777 001152 164144 MOV UNNUM,@RSCS2 ;DO NOT USE TRAP
```


2780	014742	105777	164136		INCW:	TSTB	@RSCS1	:IS BUSY CLEARED
2781	014746	100401				BMI	6\$	:FLAG CLEARED
2782	014750	104001				HLT	!CS1	:RDY NEVER CAME UP
2783	014752	013777	001102	164132	6\$:	MOV	@#OBUFSV,@RSBA	:SET UP CURRENT ADDRESS
2784	014760	013777	001202	164122		MOV	WORK2,@RSWC	:LOAD WC
2785	014766	012777	000071	164110		MOV	#71,@RSCS1	:GO READ
2786	014774	105777	164104		5\$:	TSTB	@RSCS1	:IS RDY SET?
2787	015000	100375				BPL	5\$	:WAIT FOR READ TO FINISH
2788	015002	022777	100000	164110		CMP	#100000,@RSER	:DID DCK SET?
2789	015010	001401				BEQ	+.4	:YES
2790	015012	104002				HLT	!ER	:DCK DID NOT SET
2791	015014	022777	150600	164074		CMP	#150600,@RSDS	:DID ERR SET?
2792	015022	001401				BEQ	+.4	:YES
2793	015024	104040				HLT	!DS	:ER DID NOT SET BY DCK
2794	015026	022777	144270	164050		CMP	#144270,@RSCS1	:IS CS1 CORRECT?
2795	015034	001401				BEQ	+.4	:YES
2796	015036	104044				HLT	!DS!DA	
2797	015040	017700	164042			MOV	@RSCS2,BAD	:GET CS2
2798	015044	013701	001152			MOV	UNNUM,GOOD	:GET UNIT #
2799	015050	052701	000100			BIS	#100,GOOD	:SET IR
2800	015054	020100				CMP	GOOD,BAD	:IS CS2 CORRECT?
2801	015056	001401				BEQ	+.4	:YES
2802	015060	104000				HLT		
2803	015062	012701	177700			MOV	#177700,GOOD	:NO
2804	015066	017700	164016		1\$:	MOV	@RSWC,BAD	:DID TRANSFER STOP AT END OF SECTOR?
2805	015072	020100				CMP	GOOD,BAD	
2806	015074	001401				BEQ	+.4	:YES
2807	015076	104000				HLT		:NO
2808	015100	012701	027530			MOV	#OUTBUF,GOOD	:GET BA
2809	015104	022737	000004	001162		CMP	#4,RS04DT	:LA DISK?
2810	015112	001003				BNE	7\$	:NO
2811	015114	062701	000100			ADD	#100,GOOD	:YES
2812	015120	000410				BR	3\$	:CONTINUE
2813	015122	005737	001162		7\$:	TST	RS04DT	:RS04?
2814	015126	001003				BNE	2\$	:YES
2815	015130	062701	000200			ADD	#200,GOOD	:RS03
2816	015134	000402				BR	3\$	
2817	015136	062701	000400		2\$:	ADD	#400,GOOD	:GET CORRECT ANS FOR BA
2818	015142	017700	163744		3\$:	MOV	@RSBA,BAD	:GET BA
2819	015146	020001				CMP	BAD,GOOD	:IS BA CORRECT?
2820	015150	001401				BEQ	+.4	:YES
2821	015152	104000				HLT		:NO
2822	015154	104414				CLRDK		:CLEAR ALL RS REG
2823	015156	005777	163736			TST	@RSER	:DID DCK CLEAR?
2824	015162	001401				BEQ	+.4	:YES
2825	015164	104002				HLT	!ER	:DCK DID NOT CLEAR WITH CLEAR
2826	015166	012777	177500	163714		MOV	#177500,@RSWC	:CLEAR DCK ON
2827	015174	013777	001102	163710		MOV	@#OBUFSV,@RSBA	:DRIVE BY WRITING
2828	015202	012777	000061	163674		MOV	#61,@RSCS1	:GOOD DATA
2829	015210	105777	163670		4\$:	TSTB	@RSCS1	:ON DRIVE
2830	015214	100375				BPL	4\$	


```
2831 ;TEST THE ABILITY OF THE DISK CONTROL TO
2832 ;INCREMENT THE TRACK REGISTER.
2833
2834 ;A ONE WORD WRITE WILL BE EXECUTED
2835 ;RSDA=7777 RSWC = -1
2836 ;AT THE COMPLETION OF THE WRITE RSDA = 10000
2837 ;*****
2838 ;TEST 100 TEST DISK ADDRESS REGISTER
2839 ;*****
2840 015216 104400 TST100: SCOPE
2841
2842 015220 104414 DKADR: CLRDK ;CLEAR ALL RS REG
2843 015222 012777 177777 163660 MOV #177777,@RSWC ;SET WORD COUNT TO -1
2844 015230 013777 001102 163654 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
2845 015236 012777 007777 163650 MOV #77777,@RSDA ;SET RSDA TO ALL ONES
2846 015244 012777 000061 163632 MOV #61,@RSCS1 ;GO WRITE ONE WORD
2847 015252 004737 026744 JSR PC,WAITRY ;WAIT FOR READY
2848 015256 104001 HLT !CS1 ;RDY DID NOT COME UP
2849 015260 027727 163630 010000 5$: CMP @RSDA,#10000 ;DOES RSDA=0
2850 015266 001401 BEQ .+4 ;RSDA OK
2851 015270 104004 HLT !DA ;DA DID NOT INCREMENT
2852
2853 ;*****
2854 ;TEST 101 TEST IAE ERROR
2855 ;*****
2856 015272 104400 TST101: SCOPE
2857
2858 ;IAE ERROR SHOULD SET ERR,ATA AND SC BITS
2859
2860 015274 104414 IAERR: CLRDK ;CLEAR ALL RS REG
2861 015276 012777 177777 163604 MOV #177777,@RSWC ;SET WC TO -1
2862 015304 013777 001102 163600 MOV @#OBUFSV,@RSBA ;SET UP BUS ADDRESS
2863 015312 012777 017777 163574 MOV #17777,@RSDA ;SET DA TO RECEIVE ERROR
2864 015320 012777 000061 163556 MOV #61,@RSCS1 ;GO WRITE ONE WD
2865 015326 105777 163552 7$: TSTB @RSCS1 ;TEST FOR ERR OR RDY
2866 015332 100401 BMI .+4 ;OK CONT.
2867 015334 000774 BR 7$ ;WAIT
2868 015336 022777 002000 163554 CMP #2000,@RSER ;DID IAE SET?
2869 015344 001401 BEQ .+4 ;YES
2870 015346 104002 HLT !ER ;IAE SHOULD BE SET
2871 015350 022777 150600 163540 CMP #150600,@RSDS ;DID ERR SET?
2872 015356 001401 BEQ .+4 ;YES
2873 015360 104140 HLT !DS!AS ;ERR SHOULD BE SET
2874 015362 022777 144260 163514 CMP #144260,@RSCS1 ;DID SC SET?
2875 015370 001401 BEQ .+4 ;YES
2876 015372 104001 HLT !CS1 ;SC SHOULD BE SET
2877 015374 104414 CLRDK ;CLEAR ALL RS REG
2878 015376 005777 163516 TST @RSER ;CLR ERRORS?
2879 015402 001401 BEQ .+4 ;YES
2880 015404 104002 HLT !ER ;ERR DID NOT CLR WITH 40 IN CS2
```



```
2881 ;IN THIS ROUTINE THE PROGRAM WILL GENERATE A
2882 ;NON-EXISTENT DISK ERROR
2883
2884 ;*****
2885 ;TEST 102 TEST FOR NON-EXISTENT DISK ERROR
2886 ;*****
2887 TST102: SCOPE
2888
2889 015406 104400 NEDTST: CLRDK ;CLEAR ALL RS REG
2890 015410 104414 000401 001176 MOV #401,WORK ;SET UP FOR N.E.D. NUMBER
2891 015412 012737 000401 001176 CLR GOOD ;LOOK FOR NON EXISTENT DRIVES
2892 015420 005001 001176 001154 1$: BIT WORK,UNITSV ;ON THE SYSTEM
2893 015422 033737 001176 001154 1$: BEQ 3$ ;FOUND NON EXISTENT DRIVE
2894 015430 001405 001176 001154 1$: INC GOOD ;CONTAINS UNIT #
2895 015432 005201 001176 001154 1$: ROL WORK ;KEEP LOOKING
2896 015434 006137 001176 001154 1$: BCS NEDDON ;COULD NOT FIND ANY NON EXISTENT DRIVES
2897 015442 103452 001176 001154 1$: BR 1$ ;LOOK FOR NED
2898 015444 000767 001176 001154 1$: 3$: MOV GOOD,@RSCS2 ;LOAD NED IN CS2
2899 015446 010177 163436 001176 001154 1$: CLR @RSDA ;WRITE DRIVE REG
2900 015450 005077 163440 001176 001154 1$: TST @RSDA ;DID ANY BITS SET IN RSER?
2901 015454 005777 163440 001176 001154 1$: BEQ .+4 ;NO
2902 015460 001401 163416 001176 001154 1$: HLT !DS ;WHY DID RSER CHANGE?
2903 015462 104040 001176 001154 1$: MOV @RSCS2,BAD ;GET CS2
2904 015464 017700 163416 001176 001154 1$: BIS #10100,GOOD ;SET NED AND IR
2905 015470 052701 010100 001176 001154 1$: CMP GOOD,BAD ;IS CS2 CORRECT?
2906 015474 020100 001176 001154 1$: BEQ .+4 ;YES
2907 015476 001401 160200 163374 001176 001154 1$: HLT ;GOOD=CORRECT CS2 BAD=CS2
2908 015500 104000 160200 163374 001176 001154 1$: CMP #160200,@RSCS1 ;IS CS1 CORRECT?
2909 015502 022777 160200 163374 001176 001154 1$: BEQ .+4 ;YES
2910 015510 001401 163402 001176 001154 1$: HLT !CS2 ;TRE SHOULD SET BY NED ERROR
2911 015512 104200 163402 001176 001154 1$: TST @RSAS ;DID ANY BITS SET?
2912 015514 005777 163402 001176 001154 1$: BEQ .+4 ;NO
2913 015520 001401 163402 001176 001154 1$: HLT !AS ;WHY DID AT BITS SET?
2914 015522 104100 163402 001176 001154 1$: MOV #100,@RSCS1B ;CLEAR TRE
2915 015524 112777 000100 163406 001176 001154 1$: BIT #NED,@RSCS2 ;DID NED CLEAR
2916 015532 032777 010000 163346 001176 001154 1$: BEQ .+4 ;YES
2917 015540 001401 163344 001176 001154 1$: HLT !CS2 ;NED DID NOT CLEAR
2918 015542 104200 163344 001176 001154 1$: MOV @RSDA,WORK ;READ DRIVE REG
2919 015544 017737 163344 001176 001154 1$: BIT #NED,@RSCS2 ;DID NED SET?
2920 015552 032777 010000 163326 001176 001154 1$: BNE .+4 ;NED DID NOT SET
2921 015560 001001 010000 001160 001154 1$: HLT !DS ;GET OUT
2922 015562 104040 010000 001160 001154 1$: BR NNDD ;DID THIS TYPED BEFORE?
2923 015564 000431 010000 001160 001154 1$: BIT #BIT12,ONCEE ;YES
2924 015566 032737 010000 001160 001154 1$: BNE NNDD ;ASCIZ <15><12>'COULD NOT FIND A NON-EXISTENT DRIVE'
2925 015574 001025 015602 001160 001154 1$: TYPE .+2 ;SET TYPED FLAG
2926 015576 104402 010000 001160 001154 1$: BIS #BIT12,ONCEE
```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 68
TST103 TEST THAT DAO IN RSER AND LBT IN RSDS DO SET

SEQ 0067

```
2927 :*****
2928 :TEST 103 TEST THAT DAO IN RSER AND LBT IN RSDS DO SET
2929 :*****
2930 015656 104400 TST103: SCOPE
2931
2932 015660 104414 DAOTST: CLRDK ;CLEAR ALL RS REG
2933 015662 022737 000004 001162 CMP #4,RS04DT ;RS03LA?
2934 015670 001004 BNE 3$ ;NO
2935 015672 012777 177737 163210 MOV #-41,ARSWC ;LOAD WORD COUNT
2936 015700 000411 BR 1$ ;CONT
2937 015702 012777 177577 163200 3$: MOV #-201,ARSWC ;LOAD WC FOR RS04
2938 015710 005737 001162 TST RS04DT ;IS THIS A RS04?
2939 015714 001003 BNE 1$ ;YES
2940 015716 012777 177677 163164 MOV #-101,ARSWC ;NO
2941 015724 012777 007777 163162 1$: MOV #7777,ARSDA ;SET RSDA=TO ALL ONES
2942 015732 013777 001102 163152 2$: MOV @#OBUFSV,ARSDA ;CURRENT ADDRESS=OUTBUF
2943 015740 012777 000061 163136 MOV #61,ARSCS1 ;WRITE
2944 015746 004737 026744 JSR PC,WAITRY ;WAIT FOR READY
2945 015752 104001 HLT !CS1 ;RDY DID NOT SET
2946 015754 022777 001000 163136 CMP #1000,ARSER ;DID DAO SET?
2947 015762 001401 BEQ .+4 ;YES
2948 015764 104002 HLT !ER ;DAO DID NOT SET
2949 015766 022777 152600 163122 CMP #152600,ARSDS ;DID LBT SET?
2950 015774 001401 BEQ .+4 ;YES
2951 015776 104040 HLT !DS ;LBT DID NOT SET
2952 016000 005777 163100 TST @ARSCS1 ;IS ERROR FLAG SET
2953 016004 100401 BMI .+4 ;ERROR IS SET
2954 016006 104001 HLT !CS1 ;SC DID NOT SET
2955 016010 104414 CLRDK ;CLEAR ALL RS REG
2956 016012 022777 010600 163076 CMP #10600,ARSDS ;DID ATA +LBT CLEAR
2957 016020 001401 BEQ .+4 ;YES
2958 016022 104040 HLT !DS ;ATA DID NOT CLEAR BY CLR BIT
2959 016024 005777 163070 TST @ARSER ;DID DAO CLEAR?
2960 016030 001401 BEQ .+4 ;YES
2961 016032 104002 HLT !ER ;DAO DID NOT CLEAR WITH CLEAR
```



```
2962
2963
2964
2965 016034 104400
2966
2967 016036 104414
2968 016040 022737 000004 001162
2969 016046 001004
2970 016050 012777 177741 163032
2971 016056 000411
2972 016060 012777 177601 163022 3$:
2973 016066 005737 001162
2974 016072 001003
2975 016074 012777 177701 163006
2976 016102 012777 007777 163004 1$:
2977 016110 013777 001102 162774 2$:
2978 016116 012777 000061 162760
2979 016124 004737 026744
2980 016130 104001
2981 016132 005777 162762
2982 016136 001401
2983 016140 104002
2984 016142 022777 012600 162746
2985 016150 001401
2986 016152 104040
2987 016154 005777 162724
2988 016160 100001
2989 016162 104001
2990 016164 104414
2991 016166 022777 010600 162722
2992 016174 001401
2993 016176 104040

*****
:TEST 104 TEST THAT LBT DOES SET AND DAO DOES NOT
*****
TST104: SCOPE

DAOTT: CLRDK ;CLEAR ALL RS REG
CMP #4,RS04DT ;RS03LA?
BNE 3$ ;NO
MOV #-37,ARSWC ;LOAD WORD COUNT
BR 1$ ;CONT
MOV #-177,ARSWC ;LOAD WC FOR RS04
TST RS04DT ;IS THIS A RS04?
BNE 1$ ;YES
MOV #-77,ARSWC ;NO
MOV #7777,ARSDA ;SET RSDA=TO ALL ONES
MOV @#OBUFSV,AR$BA ;CURRENT ADDRESS=OUTBUF
MOV #61,ARSCS1 ;WRITE
JSR PC,WAITRY ;WAIT FOR READY
HLT !CS1 ;RDY DID NOT SET
TST @R$ER ;ANY ERRORS?
BEQ .+4 ;NO
HLT !ER ;YES
CMP #12600,ARSDS ;DID LBT SET?
BEQ .+4 ;YES
HLT !DS ;LBT DID NOT SET
TST @RSCS1 ;IS ERROR FLAG SET
BPL .+4 ;NO
HLT !CS1 ;ERROR
CLRDK ;CLEAR ALL RS REG
CMP #10600,ARSDS ;DID LBT CLEAR
BEQ .+4 ;YES
HLT !DS ;ATA DID NOT CLEAR BY CLR BIT
```


CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 70
TST105 EXECUTE FUNCTION WITH ERROR BITS SET

SEQ 0069

```
2994  
2995  
2996  
2997 016200 104400  
2998  
2999 016202 104414  
3000 016204 012777 177017 162706  
3001 016212 017700 162704  
3002 016216 013701 001156  
3003 016222 042701 177400  
3004 016226 020100  
3005 016230 001401  
3006 016232 104100  
3007 016234 022777 104200 162642  
3008 016242 001401  
3009 016244 104040  
3010 016246 013777 001156 162646  
3011 016254 005777 162642  
3012 016260 001401  
3013 016262 104100  
3014  
3015 016264 022777 004200 162612  
3016 016272 001401  
3017 016274 104002  
3018 016276 012737 177777 027530  
3019 016304 013777 001102 162600  
3020 016312 012777 177777 162570  
3021 016320 012777 000071 162556  
3022 016326 032777 000001 162550  
3023 016334 001401  
3024 016336 104001  
3025 016340 105777 162540  
3026 016344 100375  
3027 016346 022777 144270 162530  
3028 016354 001401  
3029 016356 104002
```

```
*****  
:TEST 105 EXECUTE FUNCTION WITH ERROR BITS SET  
*****  
TST105: SCOPE  
  
ERTST: CLRDK ;CLEAR ALL RS REG  
MOV #177017,@RSER ;LOAD ER  
MOV @RSAS,BAD ;GET AS REG  
MOV UNCMP,GOOD ;GET UNIT ATA BIT  
BIC #177400,GOOD ;CLEAR JUNK  
CMP GOOD,BAD ;IS AS REG CORRECT?  
BEQ .+4 ;YES  
HLT !AS ;AS BIT SHOULD BE SET  
CMP #104200,@RSCS1 ;DID ERRS SET IN CS1?  
BEQ .+4 ;YES  
HLT !DS ;CS1 SHOULD =104200  
MOV UNCMP,@RSAS ;CLEAR ATA BIT  
TST @RSAS ;DID IT CLEAR?  
BEQ .+4 ;YES  
HLT !AS ;COULD NOT CLEAR AS BIT  
;BY LOADING A 1 INTO IT  
CMP #4200,@RSCS1 ;DID SC CLEAR BY  
BEQ .+4 ;CLEARING ATA  
HLT !ER ;NO  
MOV #177777,OUTBUF ;DATA TO BE XFERED  
MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS  
MOV #-1,@RSWC ;LOAD WC WITH -1  
MOV #71,@RSCS1 ;DO READ FUNCTION  
BIT #1,@RSCS1 ;DID GO BIT CLEAR  
BEQ .+4 ;YES  
HLT !CS1 ;GO BIT SHOULD BE CLEARED  
1$: TSTB @RSCS1 ;WAIT FOR READY  
BPL 1$ ;WAIT  
CMP #144270,@RSCS1 ;DID ERRS CLEAR BY SETTING GO BIT?  
BEQ .+4 ;YES  
HLT !ER ;NO
```


CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 71
TST105 EXECUTE FUNCTION WITH ERROR BITS SET

SEQ 0070

3030	016360	017700	162522	MOV	@RSCS2,BAD	;GET CS2
3031	016364	012701	001100	MOV	#1100,GOOD	;GET CORRECT ANS
3032	016370	053701	001152	BIS	UNNUM,GOOD	;GET UNIT #
3033	016374	020100		CMP	GOOD,BAD	;IS CS2 CORRECT?
3034	016376	001401		BEQ	.+4	;YES
3035	016400	104000		HLT		;GOOD = WHAT CS2 SHOULD =
3036	016402	022777	150600 162506	CMP	#150600,@RSDS	;DID ERR BITS SET?
3037	016410	001401		BEQ	.+4	;NO
3038	016412	104040		HLT	!DS	;ERR BIT SHOULD BE 1
3039	016414	022777	177777 162466	CMP	#-1,@RSWC	;DID WC MOVE?
3040	016422	001401		BEQ	.+4	;NO
3041	016424	104010		HLT	!WC	;WC SHOULD = 1777777
3042	016426	005777	162462	TST	@RSDA	;DID DA MOV
3043	016432	001401		BEQ	.+4	;NO
3044	016434	104004		HLT	!DA	;DA SHOULD =0
3045	016436	023777	001102 162446	CMP	@#OBUFSV,@RSBA	;DID BA MOVE
3046	016444	001401		BEQ	.+4	;NO
3047	016446	104020		HLT	!BA	;BA MOVED
3048	016450	033777	001156 162444	BIT	UNCMP,@RSAS	;AS SHOULD BE SET
3049	016456	001001		BNE	.+4	;IS IT?
3050	016460	104100		HLT	!AS	;NO
3051	016462	022777	177017 162430	CMP	#177017,@RSER	;DID ER CHANGE?
3052	016470	001401		BEQ	.+4	;NO
3053	016472	104002		HLT	!ER	;ER SHOULD NOT CHANGE


```

3054 ;*****
3055 ;TEST 106 PAT AND MCPE TEST
3056 ;*****
3057 016474 104400 TST106: SCOPE
3058
3059 016476 104414 PATST: CLRDK ;CLEAR ALL RS REG
3060 016500 052777 000020 162400 BIS #BIT4,@RSCS2 ;SET PAT
3061 016506 005777 162416 TST @RSMR ;READ DRIVE REG
3062 016512 017700 162370 MOV @RSCS2,BAD ;GET CS2
3063 016516 012701 000120 MOV #120,GOOD ;MDPE SHOULD
3064 016522 053701 001152 BIS UNNUM,GOOD ;NOT SET
3065 016526 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
3066 016530 001401 BEQ .+4 ;YES
3067 016532 104000 HLT ;BAD = CS2 GOOD = CORRECT ANS
3068 016534 012777 000010 162366 MOV #10,@RSMR ;CAUSE PAR TO SET IN RSER
3069 016542 022777 000010 162350 CMP #10,@RSER ;DID PAR SET?
3070 016550 001401 BEQ .+4 ;YES
3071 016552 104140 HLT !AS!DS
3072 016554 017700 162342 MOV @RSAS,BAD ;GET AS REG
3073 016560 013701 001156 MOV UNCMP,GOOD ;GET UNIT ATA BIT
3074 016564 042701 177400 BIC #177400,GOOD ;CLEAR JUNK
3075 016570 020100 CMP GOOD,BAD ;IS AS REG CORRECT?
3076 016572 001401 BEQ .+4 ;YES
3077 016574 104100 HLT !AS ;AS BIT SHOULD BE SET
3078 016576 022777 104200 162300 CMP #104200,@RSCS1 ;DID ERRS SET IN CS1?
3079 016604 001401 BEQ .+4 ;YES
3080 016606 104040 HLT !DS ;CS1 SHOULD =104200
3081 016610 104414 CLRDK ;CLEAR RS REG
3082 016612 022777 004200 162264 CMP #4200,@RSCS1 ;IS CS1 CORRECT?
3083 016620 001401 BEQ .+4 ;CLEARING ATA
3084 016622 104002 HLT !ER ;NO
3085 016624 017700 162256 MOV @RSCS2,BAD ;CHECK TO SEE
3086 016630 013701 001152 MOV UNNUM,GOOD ;IF PAT CLEARS
3087 016634 052701 000100 BIS #100,GOOD
3088 016640 020100 CMP GOOD,BAD
3089 016642 001401 BEQ .+4
3090 016644 104000 HLT ;PAT DID NOT CLEAR

```



```

3091 ;*****
3092 ;TEST 107 SET PAT BIT AND LOAD FUNCTION
3093 ;*****
3094 016646 104400 TST107: SCOPE
3095
3096 016650 104414 SETPAT: CLRDK ;CLEAR ALL REG
3097 016652 052777 000010 162226 BIS #BA1,@RSCS2 ;SET BAI
3098 016660 012737 177777 027530 MOV #177777,OUTBUF ;DATA TO BE XFERED
3099 016666 013777 001102 162216 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
3100 016674 012777 177000 162206 MOV #-1000,@RSWC ;LOAD WC WITH -1
3101 016702 052777 000020 162176 BIS #BIT4,@RSCS2 ;SET PAT BIT
3102 016710 012777 000071 162166 MOV #71,@RSCS1 ;DO READ FUNCTION
3103 016716 105777 162162 1$: TSTB @RSCS1 ;WAIT FOR READY
3104 016722 100375 BPL 1$ ;WAIT
3105 016724 022777 144270 162152 CMP #144270,@RSCS1 ;DID CS1 GET LOADED?
3106 016732 001401 BEQ .+4 ;NO
3107 016734 104040 HLT !DS ;IT SHOULD NOT
3108 016736 023777 001102 162146 CMP @#OBUFSV,@RSBA ;DID BA MOVE?
3109 016744 001401 BEQ .+4 ;NO
3110 016746 104020 HLT !BA ;YES
3111 016750 022777 177000 162132 CMP #-1000,@RSWC ;DID WC MOVE?
3112 016756 001401 BEQ .+4 ;NO
3113 016760 104010 HLT !WC ;YES WHY?
3114
3115 ;*****
3116 ;TEST 110 DO FUNCTION THEN SET PAT BIT
3117 ;*****
3118 016762 104400 TST110: SCOPE
3119 016764 104414 FUNDO: CLRDK ;CLEAR ALL REG
3120 016766 052777 000010 162112 BIS #BA1,@RSCS2 ;SET BAI
3121 016774 012737 177777 027530 MOV #177777,OUTBUF ;DATA TO BE XFERED
3122 017002 013777 001102 162102 MOV @#OBUFSV,@RSBA ;SET UP CURRENT ADDRESS
3123 017010 012777 177000 162072 MOV #-1000,@RSWC ;LOAD WC WITH -1
3124 017016 012777 000071 162060 3$: MOV #71,@RSCS1 ;DO A READ
3125 017024 100774 BMI 3$ ;WAIT FOR BUSY
3126 017026 052777 000020 162052 BIS #BIT4,@RSCS2 ;SET PAT
3127 017034 105777 162044 2$: TSTB @RSCS1 ;WAIT FOR READY
3128 017040 100375 BPL 2$
3129 017042 022777 144270 162034 CMP #144270,@RSCS1 ;DID MCPE SET?
3130 017050 001401 BEQ .+4 ;NO
3131 017052 104002 HLT !ER ;YES
3132 017054 017700 162026 MOV @RSCS2,BAD ;GET CS2
3133 017060 012701 000730 MOV #730,GOOD ;GET CORRECT ANG
3134 017064 053701 001152 BIS UNNUM,GOOD ;GET UNIT #
3135 017070 020100 CMP GOOD,BAD ;IS CS2 CORRECT?
3136 017072 001401 BEQ .+4 ;YES
3137 017074 104000 HLT ;GOOD = WHAT CS2 SHOULD =

```



```

3138 017076 022777 010600 162012      CMP      #10600, @RSDS      ;DID ERR BITS SET?
3139 017104 001401                      BEQ      .+4              ;NO
3140 017106 104040                      HLT      !DS              ;ERR BIT SHOULD BE 1
3141 017110 005777 162004              TST      @RSER              ;IS ER CLEAR?
3142 017114 001401                      BEQ      .+4              ;YES
3143 017116 104044                      HLT      !DS!DA           ;NO ERRORS SHOULD BE SET
3144 017120 104414                      CLRDK                     ;CLEAR ALL RS REG
3145 017122 022777 004200 161754      CMP      #4200, @RSCS1     ;IS CS1 CORRECT?
3146 017130 001401                      BEQ      .+4              ;YES
3147 017132 104040                      HLT      !DS
3148                                     ;*****
3149                                     ;TEST 111                      TEST PAR BY SETTING PAT
3150                                     ;*****
3151 017134 104400      TST111: SCOPE
3152
3153 017136 104414      PATTST: CLRDK      ;CLEAR ALL RS REG
3154 017140 052777 000010 161740      BIS      #BAI, @RSCS2     ;SET BAI
3155 017146 012737 177777 027530      MOV      #177777, OUTBUF ;DATA TO BE XFERED
3156 017154 013777 001102 161730      MOV      @#OBUFSV, @RSBA ;SET UP CURRENT ADDRESS
3157 017162 012777 177000 161720      MOV      #-1000, @RSWC   ;LOAD WC WITH -1
3158 017170 012777 000061 161706      MOV      #61, @RSCS1     ;DO WRITE FUNCTION
3159 017176 105777 161702      1$:      TSTB      @RSCS1         ;WAIT FOR READY
3160 017202 100775                      BMI      1$              ;WAIT
3161 017204 052777 000020 161674      BIS      #BIT4, @RSCS2   ;SET PAT
3162 017212 105777 161666      2$:      TSTB      @RSCS1         ;WAIT FOR READY
3163 017216 100375                      BPL      2$
3164 017220 022777 144260 161656      CMP      #144260, @RSCS1 ;DID MCPE SET?
3165 017226 001401                      BEQ      .+4              ;NO
3166 017230 104002                      HLT      !ER              ;YES
3167 017232 017700 161650      MOV      @RSCS2, BAD      ;GET CS2
3168 017236 012701 000230      MOV      #230, GOOD      ;GET CORRECT ANS-- DO NOT CHECK IR - REASON 50 CYCLE
3169 017242 053701 001152      BIS      UNNUM, GOOD      ;GET UNIT #
3170 017246 042700 000100      BIC      #BIT6, BAD      ;CLEAR IR BIT FOR CS2 COMPARE
3171 017252 020100      CMP      GOOD, BAD      ;IS CS2 CORRECT?
3172 017254 001401                      BEQ      .+4              ;YES
3173 017256 104000      HLT                     ;GOOD = WHAT CS2 SHOULD =
3174 017260 022777 150600 161630      CMP      #150600, @RSDS   ;DID ERR BITS SET?
3175 017266 001401                      BEQ      .+4              ;NO
3176 017270 104040                      HLT      !DS              ;ERR BIT SHOULD BE 1
3177 017272 022777 000010 161620      CMP      #10, @RSER       ;DID PAR SET?
3178 017300 001401                      BEQ      .+4              ;YES
3179 017302 104044                      HLT      !DS!DA           ;NO
3180 017304 104414                      CLRDK                     ;CLEAR ALL RS REG
3181 017306 022777 004200 161570      CMP      #4200, @RSCS1     ;IS CS1 CORRECT?
3182 017314 001401                      BEQ      .+4              ;YES
3183 017316 104040                      HLT      !DS
3184 017320 005777 161574      TST      @RSER              ;DID PAR CLEAR?
3185 017324 001401                      BEQ      .+4              ;YES
3186 017326 104040                      HLT      !DS

```


CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 75
TST112 TEST THE ABILITY TO FILL THE LAST SECTOR

SEQ 0074

```
3187 ;*****
3188 ;TEST 112 TEST THE ABILITY TO FILL THE LAST SECTOR
3189 ;*****
3190 017330 104400 TST112: SCOPE
3191
3192 017332 104414 LASTSC: CLRDK ;CLEAR ALL RS REG
3193 017334 012777 007777 161552 MOV #7777,@RSDA ;SET RSDA=TO ALL ONES
3194 017342 022737 000004 001162 CMP #4,RS04DT ;LA DISK?
3195 017350 001004 BNE 2$ ;NO
3196 017352 012777 177740 161530 MOV #-40,@RSWC ;LOAD WORD COUNT
3197 017360 000411 BR 1$ ;CONTINUE
3198 017362 012777 177700 161520 2$: MOV #-100,@RSWC ;WORD COUNT=-100
3199 017370 005737 001162 TST RS04DT ;IS THIS A RS04?
3200 017374 001403 BEQ 1$ ;NO
3201 017376 012777 177600 161504 MOV #-200,@RSWC ;YES
3202 017404 013777 001102 161500 1$: MOV @#OBUFSV,@RSBA ;CURRENT ADDRESS=OUTBUF
3203 017412 012777 000061 161464 MOV #61,@RSCS1 ;WRITE
3204 017420 004737 026744 JSR PC,WAITRY ;WAIT FOR READY
3205 017424 104001 HLT !CS1 ;RDY DID NOT SET
3206 017426 005777 161466 TST @RSER ;DID ANY ERROR BITS SET?
3207 017432 001401 BEQ .+4 ;NO
3208 017434 104002 HLT !ER ;GOT AN ERROR
3209 017436 022777 012600 161452 CMP #12600,@RSDS ;DID LBT SET?
3210 017444 001401 BEQ .+4 ;YES
3211 017446 104040 HLT !DS ;LBT DID NOT SET
3212 017450 005777 161430 TST @RSCS1 ;IS ERROR FLAG SET
3213 017454 100001 BPL .+4 ;ERROR IS SET
3214 017456 104001 HLT !CS1 ;SC DID NOT SET
3215 017460 104414 CLRDK ;CLEAR ALL RS REG
3216 017462 022777 010600 161426 CMP #10600,@RSDS ;DID ATA +LBT CLEAR
3217 017470 001401 BEQ .+4 ;YES
3218 017472 104040 HLT !DS ;ATA DID NOT CLEAR BY CLR BIT
```



```
3219 ;FILL SECTOR WITH ALL ONES.
3220 ;NOW WRITE 1ST WORD IN SECTOR
3221 ;TEST REMAINING 63 WORDS FOR 0
3222
3223 ;*****
3224 ;TEST 113 TEST FOR ZERO'S IN PARTIAL FILLED SECTOR
3225 ;*****
3226 017474 104400 TST113: SCOPE
3227 017476 104414 SECT: CLRDK ;CLEAR ALL RS REG
3228 017500 012737 177777 027530 MOV #-1,OUTBUF ;PUT -1INTO OUTBUF
3229 017506 013777 001102 161376 MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDR
3230 017514 022737 000004 001162 CMP #4,RS04DT ;RS03LA DISK?
3231 017522 001004 BNE 4$ ;NO
3232 017524 012777 177740 161356 MOV #-40,@RSWC ;LOAD WORD COUNT
3233 017532 000411 BR 5$ ;CONTINUE
3234 017534 012777 177600 161346 4$: MOV #-200,@RSWC ;LOAD WC FOR RS04
3235 017542 005737 001162 TST RS04DT ;RS04?
3236 017546 001003 BNE 5$ ;YES
3237 017550 012777 177700 161332 MOV #-100,@RSWC ;SET WORD COUNT TO -100
3238 017556 052777 000010 161322 5$: BIS #BAI,@RSCS2 ;SET BAI BIT
3239 017564 012777 000061 161312 MOV #61,@RSCS1 ;WRITE
3240 017572 105777 161306 3$: TSTB @RSCS1 ;IS RDY SET?
3241 017576 100375 BPL 3$ ;NO
3242 017600 005077 161310 CLR @RSDA ;SET DSK ADDRESS TO 0
3243 017604 012737 177777 027530 MOV #-1,OUTBUF ;PUT 177777 INTO OUTBUF
3244 017612 013777 001102 161272 MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDR
3245 017620 012777 177777 161262 MOV #-1,@RSWC ;SET WORD COUNT TO -1
3246 017626 052777 000010 161252 BIS #10,@RSCS2 ;SET BAI BIT
3247 017634 012777 000061 161242 MOV #61,@RSCS1 ;WRITE
3248 017642 105777 161236 1$: TSTB @RSCS1 ;IS RDY SET?
3249 017646 100375 BPL 1$ ;NO
3250 017650 042777 000010 161230 BIC #10,@RSCS2 ;CLEAR BAI BIT
3251 017656 005737 001162 TST RS04DT ;RS04?
3252 017662 001404 BEQ 7$ ;NO
3253 017664 012737 000200 001176 MOV #200,WORK ;YES
3254 017672 000403 BR 8$ ;CONT
3255 017674 012737 000100 001176 7$: MOV #100,WORK ;SET UP BUFFER
3256 017702 013701 001102 8$: MOV @#OBUFSV, R1 ;GET STARTING ADD OF BUF
3257 017706 012721 177777 6$: MOV #-1,(R1)+ ;LOAD FIRST WD WITH -1
3258 017712 005021 CLR (R1)+ ;LOAD REST WITH 0
3259 017714 005337 001176 DEC WORK ;DONE YET?
3260 017720 001374 BNE 6$ ;NO
3261 017722 005077 161166 CLR @RSDA ;SET DSK ADDRESS TO 0
3262 017726 013777 001102 161156 MOV @#OBUFSV, @RSBA ;SET UP CURRENT ADDR
3263 017734 022737 000004 001162 CMP #4,RS04DT ;RS03LA DISK?
3264 017742 001004 BNE 11$ ;NO
3265 017744 012777 177740 161136 MOV #-40,@RSWC ;LOAD WORD COUNT
3266 017752 000412 BR 10$ ;CONTINUE
3267 017754 005737 001162 11$: TST RS04DT ;RS04?
3268 017760 001404 BEQ 9$ ;NO
3269 017762 012777 177600 161120 MOV #-200,@RSWC ;YES
3270 017770 000403 BR 10$ ;CONT
3271 017772 012777 177700 161110 9$: MOV #-100,@RSWC ;SET WORD COUNT TO -100
3272 020000 012777 000051 161076 10$: MOV #51,@RSCS1 ;WRITE CHECK
```


3273	020006	032777	000200	161070	2\$:	BIT	#200,@RSCS1	;IS RDY SET?
3274	020014	001774				BEQ	2\$	;NO
3275	020016	013701	001152			MOV	UNNUM,GOOD	;GET UNIT #
3276	020022	052701	000100			BIS	#100,GOOD	;SET IR BIT
3277	020026	017700	161054			MOV	@RSCS2,BAD	;GET CS2
3278	020032	020100				CMP	GOOD,BAD	;IS CS2 CORRECT?
3279	020034	001401				BEQ	:+4	;YES
3280	020036	104002				HLT	!ER	;THERE WAS A WRITE CHECK ERROR

 ;TEST 114 IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDR

3285 020040 104400

TST114: SCOPE

3286						EXTTST: CLRDK		;CLEAR ALL RS REG.
3287	020042	104414				MOV	TIMES,TIMSV	;SAVE LOOP #
3288	020044	013737	022560	001164		MOV	#10,TIMES	;LOOP 10 TIMES
3289	020052	012737	000010	022560		MOV	#EXTTRP,4	;SETUP TIMEOUT TRAP
3290	020060	012737	020564	000004		MOV	#340,6	
3291	020066	012737	000340	000006		MOV	@#SR0	;IF MEMORY MANAGEMENT IS NOT
3292	020074	005737	177572			TST		;AVAILABLE THE PROGRAM WILL TRAP
3293								;AND TRANSFER TO END OF THE TEST
3294								
3295	020100	012737	020556	000004		MOV	#EXTRP,4	
3296	020106	012737	007600	172356		MOV	#7600,@#KIPAR7	;OPEN I/O REGISTERS
3297	020114	005037	172340			CLR	@#KIPAR0	;FREE FIRST 4K
3298	020120	012737	000200	172342		MOV	#200,@#KIPAR1	;ENABLE SECOND 4K
3299	020126	012737	002000	172344		MOV	#2000,@#KIPAR2	
3300	020134	012737	177406	172300		MOV	#400*256.-400+UP+RW,@#KIPDR0	;SET KIPDR0=RW UP 400 BLOCKS
3301	020142	012737	177406	172302		MOV	#400*256.-400+UP+RW,@#KIPDR1	;SET KIPDR1=RW UP 400 BLOCKS
3302	020150	012737	177406	172304		MOV	#400*256.-400+UP+RW,@#KIPDR2	;SET KIPDR2=RW UP 400 BLOCKS
3303	020156	012737	177406	172316		MOV	#400*256.-400+UP+RW,@#KIPDR7	;SET KIPDR7=RW UP 400 BLOCKS
3304	020164	012737	000001	177572		MOV	#1,@#SR0	;TURN ON MEMORY MANAGEMENT
3305	020172	012702	040000			MOV	#40000,R2	;R2 EQUALS BASE ADDR

M 6

CZRSBHO RH11-RS03/04 BFT MACY11 30A(1052) 13-OCT-80 15:16 PAGE 78
 CZRSBH.P11 13-OCT-80 15:14 TST114 IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDRESS BITS SEQ 0077

3306	020176	012712	177777		7\$:	MOV	#177777,(R2)	:INSERT PATTERN INTO 200000
3307	020202	012777	177776	160700		MOV	#-2,@RSWC	:SETUP WORDCOUNT
3308	020210	012777	177777	160674		MOV	#177777,@RSBA	:SETUP BUS ADDR
3309	020216	012777	000061	160660		MOV	#61,@RSCS1	:WRITE TWO WORDS ON DISK. RSBA
3310								:STARTS AT 177777 TO FORCE CARRY
3311								:TO SET A16
3312	020224	105777	160654			TSTB	@RSCS1	:WAIT FOR READY
3313	020230	100375				BPL	.-4	
3314	020232	005777	160646			TST	@RSCS1	
3315	020236	100002				BPL	1\$	
3316	020240	104046				HLT	!ER!DA!DS	:STATUS ERROR AFTER 2 WORD WRITE
3317	020242	000447				BR	2\$	:USING MEX0
3318	020244	022777	004660	160632	1\$:	CMP	#4660,@RSCS1	:IS CS1 CORRECT
3319	020252	001402				BEQ	3\$	:YES
3320	020254	104002				HLT	!ER	:CS2 DID NOT COMPARE
3321	020256	000441				BR	2\$	
3322	020260	005012			3\$:	CLR	(R2)	:CLEAR LOCATION 200000
3323	020262	005077	160626			CLR	@RSDA	:SETUP DA
3324	020266	012777	177777	160616		MOV	#177777,@RSBA	:SETUP BA
3325	020274	012777	177776	160606		MOV	#-2,@RSWC	:SETUP WC
3326	020302	012777	000071	160574		MOV	#71,@RSCS1	:READ TWO WORDS INTO LOCATIONS
3327								:177777 AND 200000.
3328	020310	105777	160570			TSTB	@RSCS1	:WAIT FOR READY
3329	020314	100375				BPL	.-4	
3330	020316	005777	160562			TST	@RSCS1	:ANY ERRORS?
3331	020322	100002				BPL	4\$	:BRANCH IF NO
3332	020324	104002				HLT	!ER	:ERROR OFTER READING 2 WORDS
3333	020326	000415				BR	2\$	
3334	020330	022777	004670	160546	4\$:	CMP	#4670,@RSCS1	:IS CS1 CORRECT?
3335	020336	001402				BEQ	5\$	:YES
3336	020340	104002				HLT	!ER	:CS1 DID NOT COMPARE
3337	020342	000407				BR	2\$	:READ STARTING AT 177777
3338	020344	022712	177777		5\$:	CMP	#177777,(R2)	:WAS DATA READ INTO LOCATION
3339	020350	001404				BEQ	2\$	:200000 CORRECTLY? - BRANCH IF YES
3340	020352	012701	177777			MOV	#177777,GOOD	
3341	020356	011200				MOV	(R2),BAD	
3342	020360	104000				HLT		:DATA COMPARE ERROR AT 200000
3343	020362	000240			2\$:	NOP		

N 6

CZRSBH0 RH11-RS03/04 BFT MACY11 30A(1052) 13-OCT-80 15:16 PAGE 79
 CZRSBH.P11 13-OCT-80 15:14 TST114 IF MEMORY MANAGEMENT IS AVAILABLE CHECK THE EXTENDED MEMORY ADDRESS BITS SEQ 0078

3344	020364	104414			EXTT1:	CLRDK		:CLEAR ALL REG
3345	020366	012737	004000	172344		MOV	#4000, @#KIPAR2	
3346	020374	012702	040000			MOV	#40000, R2	:R2 EQUALS THE BASE ADDR
3347	020400	012712	177777		7\$:	MOV	#177777, (R2)	:INSERT PATTERN INTO 400000
3348	020404	012777	177777	160500		MOV	#177777, @RSBA	:SETUP BUS ADDR
3349	020412	012777	177776	160470		MOV	#-2, @RSWC	:LOAD WC
3350	020420	012777	000461	160456		MOV	#461, @RSCS1	:SET BIT A16 AND WRITE
3351	020426	105777	160452			TSTB	@RSCS1	:WAIT FOR READY
3352	020432	100375				BPL	.-4	
3353	020434	005777	160444			TST	@RSCS1	:ANY ERRORS?
3354	020440	100001				BPL	4\$	:BRANCH IF NO
3355	020442	104002				HLT	!ER	:ERROR AFTER READING 2 WORDS
3356	020444	022777	005260	160432	4\$:	CMP	#5260, @RSCS1	:IS CS1 CORRECT?
3357	020452	001401				BEQ	10\$	:BRANCH IF YES
3358	020454	104002				HLT	!ER	:NO CS1 DID NOT COMPARE
3359								:READ STARTING AT 377777
3360	020456	005012			10\$:	CLR	(R2)	:CLEAR LOCATION 400000
3361								:READ TWO WORDS STARTING AT 377777
3362	020460	012777	177776	160422		MOV	#-2, @RSWC	:SETUP WC
3363	020466	005077	160422			CLR	@RSDA	:SETUP DA
3364	020472	012777	177777	160412		MOV	#177777, @RSBA	
3365	020500	012777	000471	160376		MOV	#471, @RSCS1	:CLEAR A 17 SET A16, READ
3366	020506	105777	160372			TSTB	@RSCS1	:WAIT FOR READY
3367	020512	100375				BPL	.-4	
3368	020514	005777	160364			TST	@RSCS1	:ANY ERRORS?
3369	020520	100002				BPL	11\$	:BRANCH IF NO
3370	020522	104002				HLT	!ER	:ERROR WHILE READING TWO WORDS
3371	020524	000414				BR	EXTRP	
3372	020526	022777	005270	160350	11\$:	CMP	#5270, @RSCS1	:IS CS1 CORRECT?
3373	020534	001401				BEQ	12\$	:BRANCH IF YES
3374	020536	104002				HLT	!ER	:CS1 DID NOT COMPARE
3375								:READ STARTING AT 377777
3376	020540	022712	177777		12\$:	CMP	#177777, (R2)	:WAS DATA READ INTO LOCATION 400000
3377	020544	001404				BEQ	EXTRP	:CORRECTLY? - BRANCH IF YES
3378	020546	012701	177777			MOV	#177777, GOOD	
3379	020552	011200				MOV	(R2), BAD	
3380	020554	104000				HLT		:DATA COMPARE ERROR AT 400000 IF
3381								:RECEIVED=0 - LOCATION WASN'T ACCESSED
3382	020556	005037	177572		EXTRP:	CLR	@#SRO	:TURN OFF MEMORY MANAGEMENT
3383	020562	000401				BR	EXT1	
3384	020564	000240			EXTTRP:	NOP		:UPDATE TEST NUMBERS
3385	020566	012706	000500		EXT1:	MOV	#500, SP	:RESTORE STACK
3386	020572	012737	000006	000004	MEMOUT:	MOV	#6, 4	
3387	020600	005037	000006			CLR	6	


```
3388
3389
3390
3391 020604 104400
3392
3393 020606 104414
3394 020610 012706 000500
3395 020614 012777 020666 160312
3396 020622 012777 000340 160306
3397 020630 012737 000200 177776
3398 020636 012777 000300 160240
3399 020644 012737 000500 001176
3400 020652 005337 001176
3401 020656 001375
3402 020660 104001
3403 020662 000137 020702
3404
3405 020666 022626
3406 020670 022777 004200 160206
3407 020676 001401
3408 020700 104001
3409 020702
3410
3411
3412
3413
3414 020702 104400
3415
3416 020704 012706 000500
3417 020710 013737 001164 022560
3418 020716 104414
3419 020720 012777 021000 160206
3420 020726 012777 000340 160202
3421 020734 012737 000240 177776
3422 020742 013700 177776
3423 020746 012777 177777 160134
3424 020754 013777 001102 160130
3425 020762 012777 000161 160114
3426 020770 004737 026744
3427 020774 104001
3428 020776 000403
3429
3430
3431 021000 012701 000240
3432 021004 104000
3433 021006

*****
;TEST 115 TEST PROGRAM INTERRUPT BY MOVING 300 INTO RSCS1
*****
TST115: SCOPE

QES: CLRDK ;CLEAR ALL DRIVES
MOV #500,SP ;SETUP STACK
MOV #PGTRAP,@RSVEC ;SET UP VECTOR
MOV #340,@RSVCPS ;SET TRAP PS
MOV #200,@#PS ;SET PS AT PRIORITY 4
MOV #300,@RSCS1 ;THIS SHOULD CAUSE A TRAP
MOV #500,WORK ;SETUP LOOP
1$: DEC WORK ;DEC LOOP SHOULD
BNE 1$ ;INTERRUPE BEFORE LOOP IS DONE
HLT !CS1 ;SHOULD NEVER GET HERE
JMP QESDON ;GET OUT

PGTRAP: CMP (6)+,(6)+ ;TRAP OK
CMP #4200,@RSCS1 ;DID IE CLEAR?
BEQ +4 ;YES
HLT !CS1 ;IE SHOULD BE CLEARED

QESDON:

*****
;TEST 116 TEST THAT DISK DOES NOT INTERRUPT WHEN PS IS AT 5
*****
TST116: SCOPE

INTR5: MOV #500,SP ;SETUP STACK
MOV TIMSV,TIMES ;RESTORE LOOP COUNTER
CLRDK ;CLEAR ALL RS REG
MOV #INT112,@RSVEC ;SET UP INTERRUPT VECTOR
MOV #340,@RSVCPS ;SET PRIO.
MOV #240,@#PS ;LOCK OUT ALL INTERRUPTS ABOVE
MOV @#PS,BAD ;GET PS
MOV #177777,@RSWC ;SET WORD COUNT TO -1
MOV @#OBUFSV,@RSBA ;LOAD CURRENT ADDRESS
MOV #161,@RSCS1 ;GO WRITE (INTERRUPT ENABLED)
JSR PC,WAITRY ;WAIT FOR READY
HLT !CS1 ;NO RDY NEVER CAME UP
BR INTDON ;RESTART ROUTINE

;PROCESSOR SHOULD NOT TRAP TO INT112

INT112: MOV #240,GOOD ;WHAT PS SHOULD HAVE
HLT ;GOOD = CORRECT ANS FOR PS
INTDON: ;DONE GET OUT
```



```

3434
3435
3436
3437 021006 104400
3438
3439 021010 012706 000500 INTR4: MOV #500,SP ;SETUP STACK
3440 021014 104414 CLRDK ;CLEAR ALL RS REG
3441 021016 012777 021112 160110 MOV #INT114,ARSVEC ;SET UP DISK TRAP VECTOR
3442 021024 012777 000340 160104 MOV #340,ARSVCPS ;SET PRIO.
3443 021032 012737 000200 177776 MOV #200,@#PS ;SET PROCESSOR TO PRIORITY 4
3444 021040 013700 177776 MOV @#PS,BAD ;GET PS
3445 021044 012701 000200 MOV #200,GOOD ;GET CORRECT PS
3446 021050 012777 177777 160032 MOV #177777,ARSWC ;SET WORD COUNT TO -1
3447 021056 013777 001102 160026 MOV @#OBUFSV,ARSBA ;LOAD CURRENT ADDRESS
3448 021064 012777 000161 160012 MOV #161,ARSCS1 ;WRITE (INTERRUPT ENABLE
3449 021072 005037 001176 CLR WORK
3450 021076 005237 001176 INC WORK ;WAIT FOR INTERRUPT TO OCCUR
3451 021102 001375 BNE .-4
3452 021104 104000 HLT ;GOOD=CORRECT PS BAD=WRONG PS
3453 021106 104042 HLT !ER!DS
3454 021110 000405 BR DONINT ;CONT
3455 021112 022777 004260 157764 INT114: CMP #4260,ARSCS1 ;DID IE CLEAR?
3456 021120 001401 BEQ .+4 ;YES
3457 021122 104001 HLT !CS1 ;WHY DID NOT IE CLEAR
3458 021124
3459
3460
3461
3462
3463 021124 104400
3464
3465 021126 012706 000500 ERINT: MOV #500,SP ;SETUP STACK
3466 021132 012737 000200 177776 MOV #200,@#PS ;SET PS AT PRI 4
3467 021140 012777 021216 157766 MOV #ERRINT,ARSVEC ;SET UP INTERRUPT ADD.
3468 021146 104414 CLRDK ;CLEAR ALL RS REG
3469 021150 012777 000340 157760 MOV #340,ARSVCPS ;SET PRIO.
3470 021156 012777 177777 157730 MOV #177777,ARSDA ;SET RSDA=TO ALL ONES
3471 021164 012777 177600 157716 MOV #177600,ARSWC ;WORD COUNT=-200
3472 021172 013777 001102 157712 MOV @#OBUFSV,ARSBA ;CURRENT ADDRESS=OUTBUF
3473 021200 012777 000161 157676 MOV #161,ARSCS1 ;WRITE
3474 021206 004737 026744 JSR PC,WAITRY ;WAIT FOR READY
3475 021212 104042 1$: HLT !ER!DS ;Y DIDN'T PGM INTERRUPT IS RDY SET?
3476 021214 000406 BR FINTST ;GET OUT
3477 021216 022777 144260 157660 ERRINT: CMP #144260,ARSCS1 ;IS CS1 RIGHT?
3478 021224 001401 BEQ .+4 ;YES
3479 021226 104042 HLT !ER!DS
3480 021230 022626 CMP (6)+,(6)+ ;CLEAR STACK
3481 021232 FINTST:

```



```
3482 *****
3483 :TEST 121 DYNAMIC FUNCTION TEST
3484 *****
3485 021232 104400 TST121: SCOPE
3486 :EXECUTE FUNCTION MODIFY UNIT # AND DO A DRIVE SEARCH
3487 :DRIVE SEARCH WILL ONLY BE DONE IF THERE ARE AT LEAST 2 DRIVES
3488 :2ND DRIVE MAY NOT BE TESTED YET SO IF THIS TEST FAILS CHECK 2ND DRIVE
3489 :BEFORE TRYING TO DEBUG THIS TEST
3490
3491 021234 104414 MODNUM: CLRDK ;CLEAR ALL RS REG
3492 021236 013737 022560 001164 MOV TIMES,TIMSV ;SAVE LOOP COUNT
3493 021244 012737 000010 022560 MOV #10,TIMES ;LOOP ONLY 10 TIMES
3494 021252 005037 001200 CLR WORK1 ;CLEAR WORK LOC.
3495 021256 005003 CLR R3
3496 021260 005004 CLR R4
3497 021262 012702 022210 MOV #DVTAB,R2 ;SETUP TABLE
3498 021266 012737 000401 001176 MOV #401,WORK ;SETUP TO TEST FOR MORE DRIVES
3499 021274 033737 001176 001154 7$: BIT WORK,UNITSV ;IS DRIVE ON SYSTEM?
3500 021302 001403 BEQ 6$ ;NO
3501 021304 020437 001152 CMP R4,UNNUM ;IS THIS THE SAME DRIVE?
3502 021310 001017 BNE 8$ ;NO
3503 021312 005204 6$: INC R4 ;UPDATE 2ND UNIT #
3504 021314 000241 CLC
3505 021316 006137 001176 ROL WORK ;CHECK FOR NEXT DRIVE
3506 021322 103364 BCC 7$ ;NOT DONE YET
3507 021324 032737 000010 001200 BIT #BIT3,WORK1 ;MULTI DRIVE?
3508 021332 001016 BNE 12$ ;YES
3509 021334 013705 001152 MOV UNNUM,R5 ;LOAD UNIT NO
3510 021340 005205 INC R5 ;CHANGE IT
3511 021342 042705 177770 BIC #177770,R5 ;CLEAR JUNK
3512 021346 000410 BR 12$
3513 021350 052737 000010 001200 8$: BIS #BIT3,WORK1 ;SET FOUND MULTI DRIVE
3514 021356 010422 MOV R4,(R2)+ ;LOAD UNIT # INTO TABLE
3515 021360 005203 INC R3 ;COUNT # OF DRIVES
3516 021362 010337 001172 MOV R3,SAVEE ;SAVE IT
3517 021366 000751 BR 6$
3518 021370 012777 021504 157536 12$: MOV #TSTVEC,ARSTVEC ;SETUP INT. TRAP
3519 021376 012777 000340 157532 MOV #340,ARSTVCPS ;SETUP PRIO.
3520 021404 005037 027530 CLR OUTBUF ;CLR TO READ INTO
3521 021410 013777 001102 157474 MOV @#OBUFSV,ARSTBA ;SET UP CURRENT ADDRESS
3522 021416 012777 177000 157464 MOV #-1000,ARSTWC ;SET WORD COUNT
3523 021424 012777 000060 157462 1$: MOV #60,ARSTDA ;LOAD DA
3524 021432 012702 022210 MOV #DVTAB,R2 ;GET TABLE
3525 021436 012777 000161 157440 MOV #161,ARSTCS1 ;GO WRITE
3526 021444 005703 TST R3 ;MORE THEN 1 DRIVE?
3527 021446 001003 BNE 13$ ;YES
3528 021450 010577 157432 MOV R5,ARSTCS2 ;NO MODIFY UNIT #
3529 021454 000407 BR 14$
```



```
3530 021456 012277 157424      13$: MOV (R2)+, @RSCS2 ;LOAD UNIT#
3531 021462 012777 000131 157414 MOV #131, @RSCS1 ;DO SEARCH
3532 021470 005303 DEC R3 ;DONE ALL DRIVES YET?
3533 021472 001371 BNE 13$ ;NO
3534 021474 012737 000200 177776 14$: MOV #200, @#PS ;ENABLE INTERRUPTS
3535 021502 000001 WTDV: WAIT
3536 021504 013777 001152 157374 TSTVEC: MOV UNNUM, @RSCS2 ;GET 1ST DRIVE
3537 021512 017700 157366 MOV @RSCS1, BAD ;GET CS1
3538 021516 042700 100000 BIC #BIT15, BAD ;CLEAR SC
3539 021522 012701 004260 MOV #4260, GOOD ;GET CORRECT ANS
3540 021526 020100 CMP GOOD, BAD ;IS CS1 CORRECT?
3541 021530 001402 BEQ 4$ ;NO! X-FER OK
3542 021532 104000 HLT ;CS1 SHOULD = 14270 OR 4270
3543 021534 104140 HLT !DS!AS
3544 021536 005777 157346 4$: TST @RSWC ;TEST WC
3545 021542 001401 BEQ .+4 ;WORD COUNT DID OVERFLOW
3546 021544 104010 HLT !WC ;SHOULD = 0
3547 021546 013701 001152 MOV UNNUM, GOOD ;GET CORRECT
3548 021552 052701 000100 BIS #100, GOOD ;ANS OF CS2
3549 021556 017700 157324 MOV @RSCS2, BAD ;GET CS2
3550 021562 020100 CMP GOOD, BAD ;IS CS2 CORRECT?
3551 021564 001401 BEQ .+4 ;YES
3552 021566 104000 HLT ;GOOD = CORRECT ANS FOR CS2
3553 021570 017700 157316 MOV @RSBA, BAD ;FETCH CURRENT ADDRESS
3554 021574 013701 001102 MOV @#OBUFSV, GOOD ;WHAT RSBA SHOULD EQUAL
3555 021600 062701 002000 ADD #2000, GOOD ;UPDATE IT
3556 021604 020001 CMP BAD, GOOD ;IS RSBA CORRECT
3557 021606 001401 BEQ .+4 ;YES EXECUTE CONTINUE
3558 021610 104000 HLT ;RSBA FAILED TO INCREMENT
3559 021612 022737 000004 001162 CMP #4, RS04DT ;RS03LA?
3560 021620 001006 BNE 2$ ;NO
3561 021622 022777 000100 157264 CMP #100, @RSDA ;IS DA CORRECT?
3562 021630 001420 BEQ 3$ ;OK
3563 021632 104104 HLT !DA!AS
3564 021634 000416 BR 3$ ;CONTINUE
3565 021636 005737 001162 2$: TST RS04DT ;IS THIS A RS04?
3566 021642 001006 BNE 5$ ;YES
3567 021644 022777 000070 157242 CMP #70, @RSDA ;IS DA CORRECT?
3568 021652 001407 BEQ 3$ ;YES
3569 021654 104104 HLT !DA!AS ;DA NOT CORRECT
3570 021656 000405 BR 3$ ;CONTINUE
3571 021660 022777 000064 157226 5$: CMP #64, @RSDA ;WAS RSDA INCREMENTED
3572 021666 001401 BEQ .+4 ;RSDA OK
3573 021670 104046 HLT !DA!ER!DS ;RSDA SHOULD CONTAIN A 64
3574 021672 012777 040000 157204 3$: MOV #TRE, @RSCS1 ;CLEAR ALL ERRORS IF ANY
3575 021700 032737 000010 001200 BIT #BIT3, WORK1 ;MULTI DRIVE?
3576 021706 001461 BEQ WTDV1 ;NO
3577 021710 012777 021734 157216 1$: MOV #TTVEC, @RSVEC ;SETUP INT FOR NEXT DRIVE
3578 021716 012716 021732 MOV #WTDV2, (SP) ;GET WAIT
3579 021722 012777 000100 157154 MOV #100, @RSCS1 ;SET IE
3580 021730 000002 RTI ;RETURN
3581 021732 000001 WTDV2: WAIT
```



```
3582 ;SERVICE ROUTINE FOR SEARCH FUNCTIONS
3583 TTVEC: CLR R2 ;CLEAR UNIT #
3584 CLC
3585 MOV #401,WORK
3586 1$: BIT WORK,ARSAS ;DID THIS DRIVE INT?
3587 BNE 2$ ;YES
3588 INC R2 ;UPDATE UNIT #
3589 CLC
3590 ROL WORK
3591 BCC 1$
3592 HLT !AS ;WHY DID WE INT WITH NO ATA???
3593 2$: MOV R2,ARSCS2 ;GET DRIVE
3594 CMP #110600,ARSDS ;DID PIP CLEAR?
3595 BEQ .+4 ;YES
3596 HLT !DS!AS ;PIP BIT DID NOT CLEAR
3597 CMP #104230,ARSCS1 ;DID SC SET?
3598 BEQ .+4 ;YES
3599 HLT !AS!DS ;SC DID NOT SET
3600 DEC SAVEE ;COUNT # OF INT
3601 BEQ WTDV1 ;DONE YET?
3602 MOV WORK,ARSAS ;CLEAR AS
3603 MOV #100,ARSCS1 ;SET IE
3604 MOV #WTDV2,(SP) ;RETURN TO WAIT
3605 RTI
3606 WTDV1: MOV #340,@#PS
3607 MOV #500,SP ;CLEAR STACK
3608 MOV RSVCP,ARSVEC ;RESTORE INT VECTOR
3609 CLR ARSVCP
3610 MODDON: SCOPE ;DONE
3611 MOV TIMSV,TIMES ;RESTORE LOOP COUNT
3612 MOV #340,@#PS ;RESTORE PS
3613 MOV #1,ICNT ;FUGE TEST NUMBERS
3614 OUT: JMP @#TRYNX ;TEST NEXT DRIVE
3615 .SBTTL $DONE - BELL AND SCOPE ROUTINE
3616
3617 DONE: SCOPE ;TERMINATIONG SCOPE FOR LOOPING
3618 ADD #1,PCNT+2 ;ADD 1 TO THE PASS COUNT
3619 ADC PCNT ;MAKE IT DOUBLE PREC.
3620 BIT #SW10,ASWR ;RING THE BELL?
3621 BNE 4$ ;NO!
3622 TYPE .,+2 ;.ASCIZ <BELL><177>
3623 4$: MOV @#42,R0 ;GET MONITOR ADDRESS
3624 BEQ $END1 ;IF NONE
3625 RESET
3626 SENDAD: JSR 7,(0) ;GO TO MONITOR
3627 240,240,240 ;SAVE ROOM FOR ACT11
3628 SEND1: JMP @#BEGIN ;RETURN
3629
3630 .TBIT: 0 ;T BIT FLAG
3631
3632 DVTAB: .BLKW 10
```


3633 .SBTTL \$TYPE - TTY TYPEOUT ROUTINE
3634
3635 ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY. THE
3636 ;CALL CAN BE IN ONE OF 3 FORMS: 1) 'TYPE ,ADR' - TYPES THE
3637 ;MESSAGE STARTING IN LOCATION 'ADR:' 2) 'TYPE ,CHAR' - TYPES
3638 ;THE ASCII 'CHAR', AND 3) 'PRINT <<15><12>'MESSAGE'>' - TYPES
3639 ;THE MESSAGE WHICH IS INLINE ASCII. THE FILLER CHARACTER WHICH IS
3640 ;TYPED AFTER A LINE FEED IS IN FILCHR AND THE NUMBER OF FILLERS
3641 ;IS IN FILCHR+1.
3642
3643 .TYPE: MOV R4,-(6) ;SAVE R4
3644 MOV R5,-(6) ;SAVE R5
3645 MOV @4(6),R5 ;GET ADDRESS TO BE TYPED
3646 BIT #177400,R5 ;IS IT A TYPED?
3647 BNE 1\$;NO
3648 MOV 4(6),R5 ;GET ADDRESS OF CHARACTER
3649 1\$: TSTB (R5) ;TERMINATOR?
3650 BEQ 2\$;GET OUT IF SO
3651 CMPB #12,(R5) ;IS THE CHAR A LINE FEED
3652 BNE 4\$;NO - GET OUT
3653 MOVB FILCHR+1,R4 ;GET THE FILL COUNT
3654 5\$: MOVB FILCHR,@TPB ;TYPE A FILLER
3655 TSTB @TPS ;DONE YET?
3656 BPL -4 ;NO - WAIT
3657 DEC R4 ;DEC COUNT
3658 BNE 5\$;LOOP UNTIL 0
3659 4\$: MOVB (R5)+,@TPB ;LOAD AND TYPE THE CHARACTER
3660 TSTB @TPS ;IS THE PRINTER READY
3661 BPL -4 ;WAIT UNTIL IT IS
3662 BR 1\$;GET THE NEXT CHARACTER
3663 2\$: MOV @4(6),-(6) ;GET ADDRESS TO BE TYPED
3664 ADD #2,6(6) ;ADD 2 TO THE ADDRESS
3665 CMP (6)+,4(6) ;IS IT .+2?
3666 BNE 3\$;NO
3667 ADD #2,R5 ;ADD 2 TO THE ADDRESS
3668 BIC #1,R5 ;BACK UP TO AN EVEN BYTE
3669 MOV R5,4(6) ;RESTORE ADDRESS
3670 3\$: MOV (6)+,R5 ;RESTORE R5
3671 MOV (6)+,R4 ;RESTORE R4
3672 RTI ;RETURN


```

3673      .SBTTL      $SCOPE - SCOPE LOOP HANDLER
3674
3675      ;THIS ROUTINE HANDLES THE ITERATIONS, LOOPING, ERROR
3676      ;LOOPING, AND THE DISPLAYING OF THE TEST NUMBER.
3677      ;"SCOPE" IS PLACED BETWEEN EACH SUBTEST IN THE TEST AND
3678      ;RECORDS THE STARTING ADDRESS OF THE SUBTEST IN "LAD:"
3679
3680      022366 104416      .SCOPE: KBDIN      ;GO CHECK FOR ^G
3681      022370 032777 000400 156430      BIT      #SW8,@SWR      ;LOOP ON SPEC. TEST?
3682      022376 001404      BEQ      1$      ;NO LOOP ON SPEC. TEST
3683      022400 127737 156422 001000      CMPB     @SWR,ICNT      ;ON RIGHT TEST? *SW7-0*
3684      022406 001453      BEQ      .OVER      ;NOT RIGHT TEST
3685      022410 032777 040000 156410 1$:      BIT      #SW14,@SWR      ;LOOP ON TEST?
3686      022416 001045      BNE      .KIT      ;LOOP ON TEST IS SET
3687      022420 000416      BR      3$      ;SKIP - NOP FOR XOR TESTER
3688      022422 013746 000004      MOV      @#4,-(6)      ;PUSH @#4 ON STACK
3689      022426 012737 022446 000004      MOV      #4$,@#4      ;SET FOR TIMEOUT
3690      022434 005737 177060      TST      @#177060      ;ERROR ON XOR?
3691      022440 012637 000004      MOV      (6)+,@#4      ;POP STACK INTO @#4
3692      022444 000422      BR      .SVLAD      ;NO ERROR - GO TO NEXT TEST
3693      022446 022626      4$:      CMP      (6)+,(6)+      ;CLEAR STACK
3694      022450 012637 000004      MOV      (6)+,@#4      ;POP STACK INTO @#4
3695      022454 000426      BR      .KIT      ;ERROR - LOOP ON TEST
3696      022456 032777 004000 156342 3$:      BIT      #SW11,@SWR      ;KILL ITERATIONS
3697      022464 001012      BNE      .SVLAD      ;YES - KILL ITERATIONS
3698      022466 105737 001001      TSTB     ICNT+1      ;FIRST ONE?
3699      022472 001404      BEQ      2$      ;BRANCH IF FIRST
3700      022474 123737 022560 001001      CMPB     TIMES,ICNT+1      ;DONE?
3701      022502 003013      BGT      .KIT      ;BRANCH IF NOT
3702      022504 112737 000001 001001 2$:      MOVB     #1,ICNT+1      ;FIRST ITERATION
3703      022512 105237 001000      .SVLAD: INCB     ICNT      ;COUNT TEST NUMBERS
3704      022516 011637 001010      MOV      (6),LAD      ;SAVE LOOP ADDRESS
3705      022522 013777 001000 156300      MOV      ICNT,@DISPLAY      ;DISPLAY TEST NO. AND ITERATION COUNT
3706      022530 000002      RTI      ;RETURN
3707
3708      022532 105237 001001      .KIT:      INCB     ICNT+1      ;INC THE ITERATION COUNT
3709      022536 013777 001000 156264 .OVER:      MOV      ICNT,@DISPLAY      ;SET UP DISPLAY
3710      022544 005737 001010      TST      LAD      ;FIRST ONE?
3711      022550 001760      BEQ      .SVLAD      ;YES
3712      022552 013716 001010      MOV      LAD,(6)      ;FUDGE RETURN ADDRESS
3713      022556 000002      RTI      ;FIXES PS
3714
3715      022560 000100      TIMES: 100      ;RUN 100 TIMES

```



```

3716      .SBTTL      $HLT - HLT ROUTINE (ERROR TYPEOUT)
3717
3718      ;THIS ROUTINE PRINTS OUT ERROR MESSAGES STARTING WITH THE
3719      ;ADDRESS OF THE 'HLT'. IT ALSO COUNTS THE NUMBER OF ERRORS
3720      ;AND HAS THE CAPABILITY OF LOOPING ON ERROR, BELL ON ERROR,
3721      ;'HALT' ON ERROR, AND INHIBIT TYPEOUTS. AN OPTIONAL ARGUMENT
3722      ;('HLT+3') WILL BE PLACED IN '.HLTCT:' FOR ADITIONAL TYPEOUTS.
3723
3724      022562 104416      .HLT:  KBDIN      ;GO CHECK FOR ^G
3725      022564 032777 002000 156234      BIT      #SW10,@SWR      ;BELL ON ERROR?
3726      022572 001402      BEQ      1$      ;NO - SKIP
3727      022574 104402 000007      TYPE      ,BELL      ;RING BELL
3728      022600 005237 001002      1$:  INC      ERRORS      ;COUNT THE NUMBER OF ERRORS
3729      022604 032777 020000 156214      BIT      #SW13,@SWR      ;SKIP TYPEOUT IF SET
3730      022612 001025      BNE      2$      ;SKIP TYPEOUTS
3731      022614 104402 022620      TYPE      ,.+2      ;.ASCIZ <15><12>
3732      022624 011637 001012      MOV      (6),HLTADR      ;PUT ADDRESS OF INSTRUCTION ON STACK
3733      022630 162737 000002 001012      SUB      #2,HLTADR      ;FUDGE ADDRESS
3734      022636 117737 156150 022720      MOVB     @HLTADR,.HLTCT      ;GET HLT ARGUEMENT
3735      022644 013746 001012      MOV      HLTADR,-(6)      ;PUT HLTADR ON STACK
3736      022650 104404      TYPEO      ;TYPE STACK IN OCTAL
3737      022652 104402 022656      TYPE      ,.+2      ;.ASCIZ ""
3738      022662 004737 026232      JSR      PC,RSREG      ;GO TO USER ERROR ROUTINE
3739      022666 005777 156134      2$:  TST      @SWR      ;HALT ON ERROR
3740      022672 100001      BPL      .+4      ;SKIP IF CONTINUE
3741      022674 000000      HALT      ;HALT ON ERROR!
3742      022676 032777 001000 156122      BIT      #SW9,@SWR      ;CHECK FOR INHIBIT LOOP ON ERROR
3743      022704 001003      BNE      3$      ;SKIP IF LOOP ON ERROR
3744      022706 105037 001001      CLRB     ICNT+1      ;CLEAR ITERATION COUNT
3745      022712 000002      RTI      ;RETURN
3746      022714 000137 022532      3$:  JMP      .KIT      ;LOOP ON TEST UNTIL NO ERRORS
3747
3748      022720 000000      .HLTCT: 0      ;HLT ARGUMENT

```



```
3749 .SBTTL          SOCAL - OCTAL TYPEOUT ROUTINE
3750
3751 ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY. IT WILL TYPE
3752 ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, OR TYPE THE
3753 ;16 BITS. IT IS CALLED VIA THE TYPOCT, TYPBIT, OR TYPOCS MACRO'S.
3754
3755 022722 012737 170101 023110 .TYPEB: MOV      #170101,.PR      ;SET BIT FLAG AND 16. CHARACTER COUNT
3756 022730 000411                BR          .PTIT           ;NOW TYPE IT IN BIT FORM
3757 022732 112737 000001 023110 .TYPEO: MOVB     #1,.PR        ;SET ZERO FILL SWITCH
3758 022740 000402                BR          .+6             ;SKIP
3759 022742 005037 023110 .TYPES: CLR          .PR            ;SUPPRESS LEADING ZERO'S
3760 022746 112737 177772 023111 MOVB     #-6,.PR+1         ;SET COUNT
3761 022754
3762 022754 010446                MOV      R4,-(6)           ;PUSH R4 ON STACK
3763 022756 010546                MOV      R5,-(6)           ;PUSH R5 ON STACK
3764 022760 016605 000010                MOV      10(6),R5      ;GET THE DATA
3765 022764 012704 023112                MOV      #.PR+2,R4     ;SET POINTER TO FIRST ASCII CHAR.
3766 022770 105014                CLRB      (4)              ;CLEAR FIRST BYTE
3767 022772 000411                BR          .PRF            ;ROTATE FIRST BIT
3768 022774 105014                .PRL:   CLRB      (4)       ;CLEAR BYTE OF CHARACTER
3769 022776 032737 000100 023110        BIT      #100,.PR     ;BIT TYPING MODE?
3770 023004 001004                BNE       .PRF            ;YES - SKIP 2 ROTATES
3771 023006 006105                ROL       R5              ;ROTATE BIT INTO C
3772 023010 106114                ROLB     (4)              ;PACK IT
3773 023012 006105                ROL       R5              ;ROTATE BIT INTO C
3774 023014 106114                ROLB     (4)              ;PACK IT
3775 023016 006105                .PRF:   ROL       R5       ;ROTATE BIT INTO C
3776 023020 106114                ROLB     (4)              ;PACK IT
3777 023022 105714                TSTB     (4)              ;IS IT ZERO?
3778 023024 001402                BEQ       .+6             ;SKIP INC
3779 023026 105237 023110                INCB     .PR        ;SET FILL SWITCH
3780 023032 105737 023110                TSTB     .PR        ;CHECK FILL SWITCH
3781 023036 001402                BEQ       .+6             ;SKIP BITSET
3782 023040 152724 000060                BISB     #'0,(4)+      ;MAKE INTO ASCII CHAR
3783 023044 105237 023111                INCB     .PR+1       ;INC COUNT
3784 023050 001351                BNE       .PRL          ;REPEAT
3785 023052 022704 023112                CMP      #.PR+2,R4     ;EMPTY BUFFER?
3786 023056 001002                BNE       .+6             ;SKIP IF NOT
3787 023060 112724 000060                MOVB     #'0,(4)+      ;LOAD 1 ZERO
3788 023064 105014                CLRB      (4)              ;NULL TERMINATOR
3789 023066 104402 023112                TYPE     .PR+2       ;TYPE IT
3790 023072 012605                MOV      (6)+,R5      ;POP STACK INTO R5
3791 023074 012604                MOV      (6)+,R4      ;POP STACK INTO R4
3792 023076 016666 000002 000004        MOV      2(6),4(6)    ;GET RID OF
3793 023104 012616                MOV      (6)+,(6)     ;DATA WORD
3794 023106 000002                RTI              ;RETURN
3795
3796 023110 000012                .PR:   .BLKW    12          ;COUNT, SWITCH, AND OUTPUT BUFFER
```



```

3797      .SBTTL      $POWER - POWER DOWN AND UP ROUTINES
3798
3799      ;THIS IS THE POWER FAIL ROUTINE WHICH WILL SAVE ALL
3800      ;THE GENERAL REGISTERS AND USER DEFINED REGISTERS THEN
3801      ;WAIT FOR POWER TO GO DOWN AND BE RESTORED.
3802      ;IF THERE ISN'T ENOUGH TIME FOR SAVING ALL THE REGISTERS,
3803      ;THE PROGRAM WILL HALT AT '.ILLUP'.
3804
3805      023134 012777 023260 000124 .POWER: MOV      #.ILLUP,@.PUVEC ;SET FOR FAST UP
3806      023142 012777 000340 000120      MOV      #340,@.PUVEC$+2 ;PRIO:7
3807      023150 010046      MOV      R0,-(6) ;PUSH R0 ON STACK
3808      023152 010146      MOV      R1,-(6) ;PUSH R1 ON STACK
3809      023154 010246      MOV      R2,-(6) ;PUSH R2 ON STACK
3810      023156 010346      MOV      R3,-(6) ;PUSH R3 ON STACK
3811      023160 010446      MOV      R4,-(6) ;PUSH R4 ON STACK
3812      023162 010546      MOV      R5,-(6) ;PUSH R5 ON STACK
3813      023164 010637 023264      MOV      SP,.SAVR6 ;SAVE SP
3814      023170 012777 023200 000070      MOV      #.POWUP,@.PUVEC ;SET UP VECTOR
3815      023176 000000      HALT      ;WAIT FOR PF
3816
3817      023200 013706 023264 .POWUP: MOV      .SAVR6,SP ;GET SP
3818      023204 005001      CLR      R1 ;WAIT LOOP FOR THE TTY
3819      023206 005201      1$: INC      R1 ;WAIT FOR THE INC
3820      023210 001376      BNE      1$ ;OF WORD
3821      023212 012605      MOV      (6)+,R5 ;POP STACK INTO R5
3822      023214 012604      MOV      (6)+,R4 ;POP STACK INTO R4
3823      023216 012603      MOV      (6)+,R3 ;POP STACK INTO R3
3824      023220 012602      MOV      (6)+,R2 ;POP STACK INTO R2
3825      023222 012601      MOV      (6)+,R1 ;POP STACK INTO R1
3826      023224 012600      MOV      (6)+,R0 ;POP STACK INTO R0
3827      023226 012737 023134 000024      MOV      #.POWER,@#24 ;SET UP THE POWER DOWN VECTOR
3828      023234 012737 000340 000026      MOV      #340,@#26 ;PRIO:7
3829      023242 104402 023246      TYPE      ..+2 ;.ASCIZ <15><12>'POWER'
3830      023256 000002      RTI      ;RETURN
3831
3832      023260 000000      .ILLUP: HALT      ;THE POWER UP SEQUENCE WAS STARTED
3833      023262 000776      BR      -2 ; BEFORE THE POWER DOWN WAS COMPLETE
3834
3835      023264 000000      .SAVR6: 0 ;PUT THE SP HERE
3836      023266 000024 000026      .PUVEC: 24,26 ;POWER UP VECTOR

```



```

3837      .SBTTL      $RDOCT - OCTAL INPUT ROUTINE
3838
3839      ;THIS ROUTINE CALLS RDLIN, INPUTS A LINE FROM THE TTY AND CONVERTS
3840      ;IT INTO AN OCTAL NUMBER WHICH IS THE FIRST WORD ON THE STACK.
3841
3842      023272 011646      .RDOCT: MOV      (6),-(6)      ;MOVE THE PC
3843      023274 016666 000004 000002      MOV      4(6),2(6)      ;MOVE THE PS
3844      023302 010146      MOV      R1,-(6)      ;PUSH R1 ON STACK
3845      023304 010246      MOV      R2,-(6)      ;PUSH R2 ON STACK
3846      023306 010346      MOV      R3,-(6)      ;PUSH R3 ON STACK
3847      023310 104412      4$: RDLIN      ;READ A LINE INTO INPUT
3848      023312 005001      CLR      R1      ;INIT DATA WORD
3849      023314 005037 027442      CLR      CTN      ;CLEAR COUNT WORD
3850      023320 012703 023546      MOV      #INPUT,R3      ;INIT POINTER
3851      023324 112302      1$: MOVB      (3)+,R2      ;GET A BYTE
3852      023326 122702 000015      CMPB      #15,R2      ;WAS IT A ,CR?
3853      023332 001421      BEQ      2$      ;GET OUT IF YES
3854      023334 122702 000060      CMPB      #'0,R2      ;CHECK FOR 0 OR GREATER
3855      023340 003024      BGT      3$      ;ERROR - LESS THAN 0
3856      023342 122702 000067      CMPB      #'7,R2      ;CHECK FOR 7 OR LESS
3857      023346 002421      BLT      3$      ;ERROR - GREATER THAN 7
3858      023350 006002      ROR      R2      ;GET
3859      023352 006002      ROR      R2      ;INTO
3860      023354 006002      ROR      R2      ;POSITION
3861      023356 006101      ROL      R1      ;FIRST BIT
3862      023360 006102      ROL      R2      ;GET
3863      023362 006101      ROL      R1      ;SECOND BIT
3864      023364 006102      ROL      R2      ;GET
3865      023366 006101      ROL      R1      ;THIRD BIT
3866      023370 005237 027442      INC      CTN      ;YES HE TYPED SOMETHING
3867      023374 000753      BR      1$      ;LOOP
3868      023376 010166 000012      2$: MOV      R1,12(6)      ;SAVE THE RESULT
3869      023402 012603      MOV      (6)+,R3      ;POP STACK INTO R3
3870      023404 012602      MOV      (6)+,R2      ;POP STACK INTO R2
3871      023406 012601      MOV      (6)+,R1      ;POP STACK INTO R1
3872      023410 000002      RTI      ;RETURN
3873
3874      023412      3$:      TYPE      ,+2      ;.ASCIZ '?'<15><12>
3875      023412 104402 023416      BR      4$      ;TRY AGAIN
3876      023422 000732

```


.SBTTL SRDLIN - TTY INPUT ROUTINE

;THIS ROUTINE INPUTS A LINE TERMINATED BY A RETURN INTO ADDRESS
;INPUT AND RETURNS A LINE FEED. THE BUFFER HAS A NULL TERMINATOR
;INSTEAD OF THE RETURN. RUBOUTS ARE HANDLED BY RETYPING
;THE LINE. BUFFER OVERFLOW ERRORS LIKE A RUBOUT.

.RDLIN: MOV R5, -(6) ;SAVE R5
1\$: MOV #INPUT, R5 ;GET ADDRESS
2\$: CMP #INPUT+16., R5 ;BUFFER FULL?
BEQ 4\$;YES - TYPE "?"
TSTB @#177560 ;WAIT FOR
BPL -4 ;A CHARACTER
MOVB @#177562, (5) ;GET CHARACTER
BICB #200, (5) ;GET RID OF JUNK
CMPB #25, (5) ;IS IT A ^U
BNE 5\$;BRANCH IF NOT
TYPE ,+2 ;.ASCIZ "'^U'<15><12>
BR 1\$;START OVER
5\$: CMPB #177, (5) ;IS IT A RUBOUT
BNE 3\$;SKIP IF NOT
4\$: TYPE ,+2 ;.ASCIZ "'?'<15><12>
BR 1\$;ZAP THE BUFFER AND LOOP
3\$: MOVB (5), #0 ;SET UP FOR TYPING
TYPE ,3\$+2 ;ECHO IT
CMPB #15, (5)+ ;CHECK FOR RETURN
BNE 2\$;LOOP IF NOT RETURN
TYPE ,12 ;TYPE A LINE FEED
MOV (6)+, R5 ;RESTORE R5
RTI ;RETURN

INPUT: .BLKB 16. ;TTY INPUT AREA
.SBTTL STRAP - TRAP HANDLER

;THIS ROUTINE DECODES A TRAP CALL AND JUMPS TO THE APROPRATE
;SUBROUTINE. THE CALL IS A 'TRAP+N' WHERE N IS A MULTIPLE OF 2.
;THE 'SET' MACRO WILL CREATE THE TABLE NEEDED. IT HAS TO
;FOLLOW THIS MACRO.

.TRAP: MOV (6), -(6) ;GET ADDRESS OF TRAP +2
SUB #2, (6) ;MAKE IT ADDRESS OF TRAP
MOV @ (6), (6) ;GET TRAP INSTRUCTION
ADD #.TRP+2-TRAP, (6) ;GET DATA AND MAKE IT AN OFFSET
.TRP: MOV @ (6)+, PC ;GO TO PROPER SUBROUTINE

.SCOPE	=	TRAP+0	(104400)
.TYPE	=	TRAP+2	(104402)
.TYPE0	=	TRAP+4	(104404)
.TYPES	=	TRAP+6	(104406)
.RDOCT	=	TRAP+10	(104410)
.RDLIN	=	TRAP+12	(104412)
.CLRDK	=	TRAP+14	(104414)
.KBDIN	=	TRAP+16	(104416)
.SUSWR	=	TRAP+20	(104420)
.CNTLU	=	TRAP+22	(104422)

3877
3878
3879
3880
3881
3882
3883
3884 023424 010546
3885 023426 012705 023546
3886 023432 022705 023566
3887 023436 001423
3888 023440 105737 177560
3889 023444 100375
3890 023446 113715 177562
3891 023452 142715 000200
3892 023456 122715 000025
3893 023462 001006
3894 023464 104402 023470
3895 023476 000753
3896 023500 122715 000177
3897 023504 001005
3898 023506
3899 023506 104402 023512
3900 023516 000743
3901 023520 111527 000000
3902 023524 104402 023522
3903 023530 122725 000015
3904 023534 001336
3905 023536 104402 000012
3906 023542 012605
3907 023544 000002
3908
3909 023546 000020
3910
3911
3912
3913
3914
3915
3916
3917 023566 011646
3918 023570 162716 000002
3919 023574 017616 000000
3920 023600 062716 117206
3921 023604 013607
3922
3923 023606 022366
3924 023610 022230
3925 023612 022732
3926 023614 022742
3927 023616 023272
3928 023620 023424
3929 023622 026214
3930 023624 027272
3931 023626 027152
3932 023630 027354


```
3933 ;ROUTINE TO ALLOW THE OPERATOR TO SET BITS
3934 ;IN THE I/O REGISTERS VIA THE SWITCH REGISTER
3935
3936 ;WORD COUNT REGISTER
3937 023632 012706 000500 SRSWC: MOV #500,SP ;SET UP STACK FOR TRAP CALL
3938 023636 104416 1$ : KBDIN ;CHECK THE WORLD
3939 023640 017777 155162 155242 : MOV @SWR,@RSC ;MOV SWR INTO WORD COUNT REG
3940 023646 017777 155236 155154 : MOV @RSC,@DISPLAY ;DISPLAY IN LIGHTS
3941 023654 000770 : BR 1$
3942
3943 ;CURRENT ADDRESS REGISTER
3944 023656 012706 000500 SRSBA: MOV #500,SP ;INIT THE STACK
3945 023662 104416 1$ : KBDIN ;CTW
3946 023664 017777 155136 155220 : MOV @SWR,@RSBA ;MOV SWR INTO CURRENT ADDR REG
3947 023672 017777 155214 155130 : MOV @RSBA,@DISPLAY ;SHOW IN LIGHTS
3948 023700 000770 : BR 1$
3949
3950 ;DISK ADDRESS REGISTER
3951 023702 012706 000500 SRSDA: MOV #500,SP ;INIT THE STACK
3952 023706 104416 1$ : KBDIN ;CTW
3953 023710 017777 155112 155176 : MOV @SWR,@RSDA ;MOV SWR INTO DISK ADDR REG
3954 023716 017777 155172 155104 : MOV @RSDA,@DISPLAY ;SHOW IN LIGHTS
3955 023724 000770 : BR 1$
3956
3957 ;DRIVE STATUS REGISTER
3958 023726 012706 000500 SRSDS: MOV #500,SP ;INIT THE STACK
3959 023732 104416 1$ : KBDIN ;CTW
3960 023734 017777 155066 155154 : MOV @SWR,@RSDS ;MOV SWR INTO DRIVE STATUS
3961 023742 017777 155150 155060 : MOV @RSDS,@DISPLAY ;SHOW IN LIGHTS
3962 023750 000770 : BR 1$
3963
3964 ;DRIVE ERROR REGISTER
3965 023752 012706 000500 SRSER: MOV #500,SP ;INIT THE STACK
3966 023756 104416 1$ : KBDIN ;CTW
3967 023760 017777 155042 155132 : MOV @SWR,@RSER ;LOAD ER REG
3968 023766 017777 155126 155034 : MOV @RSER,@DISPLAY ;DISPLAY IT IN LIGHTS
3969 023774 000770 : BR 1$ ;LOOP
3970
3971 ;WATCH LOOK AHEAD REGISTER
3972 023776 017777 155122 155024 SRSLA: MOV @RSLA,@DISPLAY ;SHOW IN LIGHTS
3973 024004 000774 : BR SRSLA
```



```

3974
3975 024006 012706 000500      ;RSCS2 REGISTER
3976 024012 104416      SRCS2: MOV #500,SP      ;INIT THE STACK
3977 024014 017777 155006 155064 1$: KBDIN      ;CTW
3978 024022 017777 155060 155000      MOV @SWR,@RSCS2      ;LOAD CS2
3979 024030 000770      MOV @RSCS2,@DISPLAY ;DISPLAY IT
3980
3981      ;RSAS REGISTER
3982 024032 012706 000500      SRAS: MOV #500,SP      ;INIT THE STACK
3983 024036 104416      1$: KBDIN      ;CTW
3984 024040 017777 154762 155054      MOV @SWR,@RSAS      ;LOAD RSAS
3985 024046 017777 155050 154754      MOV @RSAS,@DISPLAY ;DISPLAY IT
3986 024054 000770      BR 1$
3987
3988      ;RSMR REGISTER
3989 024056 012706 000500      RSMRR: MOV #500,SP      ;INIT THE STACK
3990 024062 104416      1$: KBDIN      ;CTW
3991 024064 017777 154736 155036      MOV @SWR,@RSMR      ;LOAD RSMR
3992 024072 017777 155032 154730      MOV @RSMR,@DISPLAY ;DISPLAY IT
3993 024100 000770      BR 1$
3994
3995      ;DISK CONTROL STATUS REGISTER
3996 024102 012706 000500      SRSCS1: MOV #500,SP      ;INIT THE STACK
3997 024106 104416      1$: KBDIN      ;CTW
3998 024110 012737 000340 177776      MOV #340,@#PS      ;LOCK UP INTERRUPTS
3999 024116 012777 177777 154764      MOV #177777,@RSWC      ;SET WORD COUNT -1 WORD
4000 024124 013777 001102 154760      MOV @#OBUFSV,@RSBA      ;SET UP CURRENT ADDRESS
4001 024132 017777 154670 154744      MOV @SWR,@RSCS1      ;MOV SWR INTO CONTROL REG
4002 024140 032777 000001 154736      BIT #BIT0,@RSCS1      ;IS FUNCTION BITS SET
4003 024146 001757      BEQ 1$      ;FUNCTION BITS NOT SET
4004 024150 105777 154730      2$: TSTB @RSCS1      ;TEST FOR DISK READY
4005 024154 100375      BPL 2$      ;DISK STILL NOT READY
4006 024156 000753      BR 1$      ;DISK NOT BUSY SECT NEW CR

```



```
4007 ;THIS ROUTINE GIVES THE OPERATOR THE ABILITY TO
4008 ;SELECT DA, WC, UNIT # AND DESIRED PATTERN. PATTERN = NUMBER TYPED
4009 ;WITH SW12 SET THE PROGRAM WILL LOOP ON A READ WITH LOC OUTBUF+2 AS
4010 ;THE BA ADDR. WITH BIT12 0 IN THE SWR THE PROGRAM
4011 ;WILL WRITE WITH OUTBUF AS THE BA ADDR. BAI IS ALWAYS SET
4012 ;SWITCHES 0 TO 11 WILL DETERMINE THE DA
4013 ;EXAMPLE:      TYPE UNIT # 5
4014 ;              TYPE POSITIVE (OCTAL) WC 64
4015 ;              TYPE PATTERN DESIRED 1252525
4016
4017 024160 012706 000500      TKSEL:  MOV    #500,SP      ;SET STACK
4018 024164 000240              NOP
4019 024166 104402 024172      TYPE    ..+2              ;.ASCIZ <15><12>'TYPE UNIT # ''
4020 024212 104410              RDOCT
4021 024214 012637 001152      MOV     (6)+,UNNUM
4022 024220 104402 024224      TYPE    ..+2              ;.ASCIZ <15><12>'TYPE POSITIVE (OCTAL) WC ''
4023 024260 104410              RDOCT
4024 024262 012637 001176      MOV     (6)+,WORK
4025 024266 005137 001176      COM     WORK
4026 024272 104402 024276      TYPE    ..+2              ;.ASCIZ <15><12>'TYPE PATTERN DESIRED ''
4027 024326 104410              RDOCT
4028 024330 012637 027530      MOV     (6)+,OUTBUF
4029 024334 042737 000001 027270  BIC     #BIT0,SWI      ;CLEAR THE BEENHEREBIT
4030 024342 104420              SUSWR      ;INIT SWITCHLESS
4031 024344 017737 154456 001202  TK1:  MOV     @SWR,WORK2      ;SAVE SWR
4032 024352 104414              TK2:  CLRDK      ;CLEAR ALL RS REG
4033 024354 052777 000010 154524      BIS     #BIT3,@RSCS2      ;SET BAI
4034 024362 017737 154440 001200      MOV     @SWR,WORK1      ;GET SWR FOR DSK ADDR
4035 024370 042737 170000 001200      BIC     #170000,WORK1      ;CLEAR UNIT #
4036 024376 013777 001200 154510      TKKS:  MOV     WORK1,@RSDA      ;LOAD THE DA
4037 024404 013777 001176 154476      MOV     WORK,@RSWC      ;LOAD WORD COUNT
4038 024412 032777 010000 154406      BIT     #BIT12,@SWR      ;READ?
4039 024420 001412              BEQ     WTE      ;NO
4040 024422 012777 027532 154462      MOV     #OUTBUF+2,@RSBA      ;LOAD CURRENT ADDRESS
4041 024430 012777 000071 154446      MOV     #71,@RSCS1      ;GO AND READ
4042 024436 105777 154442      WT:    TSTB     @RSCS1      ;TEST FOR READY
4043 024442 100375              BPL     .-4
4044 024444 000407              BR      SWRCHG
4045 024446 012777 027530 154436      WTE:  MOV     #OUTBUF,@RSBA
4046 024454 012777 000061 154422      MOV     #61,@RSCS1
4047 024462 000765              BR      WT
4048 024464 017777 154414 154336      SWRCHG: MOV     @RSCS1,@DISPLAY ;DISPLAY CS1
4049 024472 005777 154406              TST     @RSCS1      ;ANY ERRORS?
4050 024476 100001              BPL     1$      ;NO
4051 024500 104014              HLT     !DA!WC
4052 024502 104416              1$:   KBDIN      ;CHECK FOR NEW VALUE
4053 024504 027737 154316 001202      CMP     @SWR,WORK2      ;DID SWR CHANGE?
4054 024512 001314              BNE     TK1      ;YES
4055 024514 000716              BR      TK2      ;NO
```



```
4056 :*****
4057 :TEST 122 WRITE LOCK TEST
4058 :*****
4059 024516 104400 TST122: SCOPE
4060
4061 024520 WRTLCK:
4062 024520 104402 024524 TYPE ;..+2 ;.ASCIZ <15><12>'LOAD SW WITH UNIT # AND CONT''
4063 024564 022737 000176 001026 CMP #SWREG,SWR
4064 024572 001402 BEQ 7$ ;CHECK IF SWITCHLESS CPU
4065 024574 000000 HALT
4066 024576 000401 BR 8$ ;GO AROUND TRAP CALL
4067 024600 104422 7$: CNTLU ;GET SWREG VALUE
4068 024602 017737 154220 001152 8$: MOV @SWR,UNNUM ;GET UNIT #
4069 024610 104414 CLRDK ;CLEAR ALL REG
4070 024612 005037 027530 CLR OUTBUF ;PUT A 0 INTO DATA BUFFER
4071 024616 012777 027530 154266 MOV #OUTBUF,@RSBA ;SETUP REG TO
4072 024624 012777 007700 154262 MOV #7700,@RSDA ;TO A WRITE
4073 024632 012777 177777 154250 MOV #-1,@RSWC
4074 024640 012777 000061 154236 MOV #61,@RSCS1
4075 024646 105777 154232 6$: TSTB @RSCS1 ;WAIT FOR DONE
4076 024652 100375 BPL 6$
4077 024654 104402 024660 TYPE ;..+2 ;.ASCIZ <15><12>'SET WRITE LOCK ENABLE AND CONTINUE''
4078 024726 000000 HALT
4079 024730 012777 000000 154156 1$: MOV #0,@RSDA
4080 024736 022777 014600 154152 CMP #14600,@RSDS
4081 024744 001401 BEQ .+4
4082 024746 104044 HLT !DS!DA ;DS SHOULD=14600
4083 024750 012777 000100 154136 MOV #100,@RSDA
4084 024756 022777 010600 154132 CMP #10600,@RSDS
4085 024764 001401 BEQ .+4
4086 024766 104044 HLT !DS!DA ;DS SHOULD=10600
4087 024770 104402 024774 TYPE ;..+2 ;.ASCIZ <15><12>'SET WRT LGC SW 0 AND CONT''
4088 025030 000000 HALT
4089 025032 022777 014600 154056 2$: CMP #14600,@RSDS
4090 025040 001401 BEQ .+4
4091 025042 104044 HLT !DS!DA ;DS SHOULD=14600
4092 025044 012777 000300 154042 MOV #300,@RSDA
4093 025052 022777 010600 154036 CMP #10600,@RSDS
4094 025060 001401 BEQ .+4
4095 025062 104044 HLT !DS!DA ;DS SHOULD=10600
4096 025064 104402 025070 TYPE ;..+2 ;.ASCIZ <15><12>'SET WRT LOC SW 1 AND CONT''
4097 025124 000000 HALT
4098 025126 022777 014600 153762 3$: CMP #14600,@RSDS
4099 025134 001401 BEQ .+4
4100 025136 104044 HLT !DS!DA ;DS SHOULD=14600
4101 025140 012777 000700 153746 MOV #700,@RSDA
4102 025146 022777 010600 153742 CMP #10600,@RSDS
4103 025154 001401 BEQ .+4
4104 025156 104044 HLT !DS!DA ;DS SHOULD=10600
4105 025160 104402 025164 TYPE ;..+2 ;.ASCIZ <15><12>'SET WRT LCK SW 2 AND CONT''
4106 025220 000000 HALT
```


4107	025222	022777	014600	153666	4\$:	CMP	#14600, @RSDS	
4108	025230	001401				BEQ	.+4	
4109	025232	104044				HLT	!DS!DA	;DS SHOULD=14600
4110	025234	012777	001700	153652		MOV	#1700, @RSDA	
4111	025242	022777	010600	153646		CMP	#10600, @RSDS	
4112	025250	001401				BEQ	.+4	
4113	025252	104044				HLT	!DS!DA	;DS SHOULD=10600
4114	025254	104402	025260			TYPE	..+2	;.ASCIZ <15><12>'SET WRT LCK SW 3 AND CONT''
4115	025314	000000			HALT:	HALT		
4116	025316	022777	014600	153572		CMP	#14600, @RSDS	
4117	025324	001401				BEQ	.+4	
4118	025326	104044				HLT	!DS!DA	;DS SHOULD=14600
4119	025330	012777	003700	153556		MOV	#3700, @RSDA	
4120	025336	022777	010600	153552		CMP	#10600, @RSDS	
4121	025344	001401				BEQ	.+4	
4122	025346	104044				HLT	!DS!DA	;DS SHOULD=10600
4123	025350	104402	025354			TYPE	..+2	;.ASCIZ <15><12>'SET WRT LCK SW 4 AND CONT''
4124	025410	000000				HALT		
4125	025412	022777	014600	153476		CMP	#14600, @RSDS	
4126	025420	001401				BEQ	.+4	
4127	025422	104044				HLT	!DS!DA	;DS SHOULD=14600
4128	025424	012777	006000	153462		MOV	#6000, @RSDA	
4129	025432	022777	010600	153456		CMP	#10600, @RSDS	
4130	025440	001401				BEQ	.+4	
4131	025442	104044				HLT	!DS!DA	;DS SHOULD=10600
4132	025444	104402	025450			TYPE	..+2	;.ASCIZ <15><12>'SET WRT LCK SW 5 AND CONT''
4133	025504	000000				HALT		
4134	025506	022777	014600	153402		CMP	#14600, @RSDS	
4135	025514	001401				BEQ	.+4	
4136	025516	104044				HLT	!DS!DA	;DS SHOULD=14600
4137	025520	012777	007700	153366		MOV	#7700, @RSDA	
4138	025526	022777	014600	153362		CMP	#14600, @RSDS	
4139	025534	001401				BEQ	.+4	
4140	025536	104044				HLT	!DS!DA	;DS SHOULD=14600
4141	025540	012737	177777	027530		MOV	#-1, OUTBUF	;PUT A 1 INTO DATA BUFFER
4142	025546	012777	027530	153336		MOV	#OUTBUF, @RSBA	;SETUP REG TO
4143	025554	012777	007700	153332		MOV	#7700, @RSDA	;TO A WRITE
4144	025562	012777	177777	153320		MOV	#-1, @RSWC	
4145	025570	012777	000061	153306		MOV	#61, @RSCS1	;TRY TO WRITE
4146	025576	105777	153302		7\$:	TSTB	@RSCS1	;WAIT FOR DONE
4147	025602	100375				BPL	7\$	
4148	025604	105777	153274		5\$:	TSTB	@RSCS1	;WAIT FOR READY
4149	025610	100375				BPL	5\$	
4150	025612	022777	154600	153276		CMP	#154600, @RSDS	
4151	025620	001401				BEQ	.+4	
4152	025622	104044				HLT	!DS!DA	;DS SHOULD=154600


```
4153 025624 022777 004000 153266      CMP      #4000,@RSER
4154 025632 001401                BEQ      .+4
4155 025634 104044                HLT      !DS!DA      ;ER SHOULD=4000
4156 025636 104414                CLRDK                ;CLEAR ALL REG
4157 025640 022777 014600 153250      CMP      #14600,@RSDS      ;DID WLE CLEAR?
4158 025646 001401                BEQ      .+4                ;NO
4159 025650 104040                HLT      !DS                ;A CLEAR SHOULD NOT CLEAR WLE
4160 025652 005037 027530          CLR      OUTBUF                ;PUT A 0 INTO DATA BUFFER
4161 025656 012777 027530 153226      MOV      #OUTBUF,@RSBA      ;SETUP REG TO
4162 025664 012777 007700 153222      MOV      #7700,@RSDA      ;TO A WRITE
4163 025672 012777 177777 153210      MOV      #-1,@RSWC
4164 025700 012777 000071 153176      MOV      #71,@RSCS1      ;DO A READ TO SEE IF DISK DID
4165 025706 105777 153172      8$: TSTB      @RSCS1                ;ACTUALLY GET WRITTEN ON TO
4166 025712 100375                BPL      8$                ;WAIT FOR DONE
4167 025714 005737 027530          TST      OUTBUF                ;IS DATA STILL 0 ON THE DSK?
4168 025720 001401                BEQ      .+4                ;YES
4169 025722 104040                HLT      !DS                ;NO DSK DID GET WRITTEN ONTO WITH WLE SET
4170 025724 104402 025730          TYPE      ..+2                ;.ASCIZ <15><12>'RESET ALL W/L SWITCHES'
4171 025762 000000                HALT
4172 025764 022777 010600 153124      CMP      #10600,@RSDS      ;DID WLE CLEAR
4173 025772 001401                BEQ      .+4                ;YES
4174 025774 104040                HLT      !DS                ;NO
4175
4176      ;ROUTINE FOR FINDING MEMORY ON 'B' PORT
4177 025776 012737 026062 000004 FINDM: MOV      #MAXREF,4      ;SET UP I/O BUS TRAP
4178 026004 012737 000340 000006      MOV      #340,6      ;SET PS
4179 026012 032777 010000 153006      BIT      #BIT12,@SWR      ;INHIBIT OBUFSV FROM CHANGING?
4180 026020 001003                BNE      EXREF                ;YES
4181 026022 062737 020000 001102      ADD      #20000,OBUFSV      ;ADD 4 K
4182 026030 005777 153046      EXREF: TST      @OBUFSV      ;LEGAL LOC ? IF NO TRAPS
4183 026034 012737 000006 000004      MOV      #6,4      ;RESTORE I/O BUS TRAP
4184 026042 005037 000006          CLR      6
4185 026046 022737 177446 001102      CMP      #177446,OBUFSV      ;TEST FOR GREATER THEN 28K
4186 026054 103403                BLO      MAXRF1                ;YES
4187 026056 000137 007022          JMP      WRTSTB                ;RETRY WRITING
4188 026062 022626      MAXREF: CMP      (6)+,(6)+      ;CLEAR STACK
4189 026064 012737 000006 000004 MAXRF1: MOV      #6,4      ;RESTORE I/O BUS TRAP
4190 026072 005037 000006          CLR      6
4191 026076 012737 027530 001102      MOV      #OUTBUF,OBUFSV      ;RESTORE ORIGINAL VALUE
4192 026104 032777 020000 152714      BIT      #BIT13,@SWR      ;INHIBIT TYPEOUT?
4193 026112 001031                BNE      1$                ;YES
4194 026114 032737 020000 001160      BIT      #BIT13,ONCEE      ;DID WE TYPE THIS YET?
4195 026122 001025                BNE      1$                ;YES
4196 026124 104402 026130          TYPE      ..+2                ;.ASCIZ <15><12>'COULD NOT FIND MEMORY ON -B- PORT'<15><
4197
4198 026176 052737 020000 001160 .EVEN 1$: BIS      #BIT13,ONCEE      ;SET TYPE FLAG
4199 026204 104400                SCOPE                ;UPDATE
4200 026206 104400                SCOPE                ;TEST NUMBERS
4201 026210 000137 007612          JMP      NXM                ;CONT TESTING
4202
4203      ;CLEAR ALL DISK REGISTERS
4204 026214 012777 000040 152664 .CLRDK: MOV      #40,@RSCS2      ;CLEAR ALL DSK REG
4205 026222 013777 001152 152656      MOV      UNNUM,@RSCS2      ;GET UNIT NUMBER
4206 026230 000002                RTI
```



```
4207 ;ERROR TYPTXTOUT ROUTINE
4208
4209 026232 005737 022720 RSREG: TST .HLTCT ;SHOULD WE TYPTXT GOOD AND BAD
4210 026236 001021 BNE 8$ ;NO
4211 026240 104402 026244 TYPE ..+2 ;.ASCIZ "BAD="
4212 026252 010046 MOV BAD,-(6) ;PUT BAD ON STACK
4213 026254 104404 TYPEO ;TYPE STACK IN OCTAL
4214 026256 104402 026262 TYPE ..+2 ;.ASCIZ "GOOD="
4215 026272 010146 MOV GOOD,-(6) ;PUT GOOD ON STACK
4216 026274 104404 TYPEO ;TYPE STACK IN OCTAL
4217 026276 000137 026734 JMP PTDONE ;GET OUT
4218 026302
4219 026302 104402 026306 8$: TYPE ..+2 ;.ASCIZ "CS1="
4220 026314 017746 152564 MOV @RSCS1,-(6) ;PUT @RSCS1 ON STACK
4221 026320 104404 TYPEO ;TYPE STACK IN OCTAL
4222 026322
4223 026322 104402 026326 1$: TYPE ..+2 ;.ASCIZ "ER="
4224 026334 017746 152560 MOV @RSER,-(6) ;PUT @RSER ON STACK
4225 026340 104404 TYPEO ;TYPE STACK IN OCTAL
4226 026342
4227 026342 104402 026346 2$: TYPE ..+2 ;.ASCIZ "CS2="
4228 026354 017746 152526 MOV @RSCS2,-(6) ;PUT @RSCS2 ON STACK
4229 026360 104404 TYPEO ;TYPE STACK IN OCTAL
4230 026362 032737 000200 022720 BIT #200,.HLTCT ;TYPTXT SECOND SET ?
4231 026370 001076 BNE SEEC ;YES
4232 026372 032737 000100 022720 BIT #AS,.HLTCT ;TYPTXT ER ?
4233 026400 001410 BEQ 3$ ;NO
4234 026402 104402 026406 TYPE ..+2 ;.ASCIZ "AS="
4235 026414 017746 152502 MOV @RSAS,-(6) ;PUT @RSAS ON STACK
4236 026420 104404 TYPEO ;TYPE STACK IN OCTAL
4237 026422 032737 000020 022720 3$: BIT #BA,.HLTCT ;TYPTXT BUS ASSRESS
4238 026430 001410 BEQ 4$ ;NO
4239 026432 104402 026436 TYPE ..+2 ;.ASCIZ "BA="
4240 026444 017746 152442 MOV @RSBA,-(6) ;PUT @RSBA ON STACK
4241 026450 104404 TYPEO ;TYPE STACK IN OCTAL
4242 026452 032737 000004 022720 4$: BIT #DA,.HLTCT ;TYPTXT DA ?
4243 026460 001410 BEQ 5$ ;NO
4244 026462 104402 026466 TYPE ..+2 ;.ASCIZ "DA="
4245 026474 017746 152414 MOV @RSDA,-(6) ;PUT @RSDA ON STACK
4246 026500 104404 TYPEO ;TYPE STACK IN OCTAL
4247 026502 032737 000010 022720 5$: BIT #WC,.HLTCT ;TYPTXT WC?
4248 026510 001410 BEQ 6$ ;NO
4249 026512 104402 026516 TYPE ..+2 ;.ASCIZ "WC="
4250 026524 017746 152360 MOV @RSWC,-(6) ;PUT @RSWC ON STACK
4251 026530 104404 TYPEO ;TYPE STACK IN OCTAL
```


4252	026532	032737	000040	022720	6\$:	BIT	#DS,.HLTCT	:DRIVE STATUS
4253	026540	001475				BEQ	PTDONE	:NO
4254	026542	104402	026546			TYPE	..+2	:.ASCIZ " DS="
4255	026554	017746	152336			MOV	@RSDS,-(6)	:PUT @RSDS ON STACK
4256	026560	104404				TYPEO		:TYPE STACK IN OCTAL
4257	026562	000137	026734			JMP	PTDONE	:GET OUT
4258	026566	042737	000200	022720	SEEC:	BIC	#200,.HLTCT	:CLEAR COMMON BIT
4259	026574	032737	000240	022720		BIT	#DT,.HLTCT	:TYPTXT DRIVE TYPE?
4260	026602	001410				BEQ	9\$	:NO
4261	026604	104402	026610			TYPE	..+2	:.ASCIZ " DT="
4262	026616	017746	152310			MOV	@RSDT,-(6)	:PUT @RSDT ON STACK
4263	026622	104404				TYPEO		:TYPE STACK IN OCTAL
4264	026624	032737	000210	022720	9\$:	BIT	#DB,.HLTCT	:TYPTXT DATA BUFFER
4265	026632	001410				BEQ	10\$	:NO
4266	026634	104402	026640			TYPE	..+2	:.ASCIZ " DB="
4267	026646	017746	152254			MOV	@RSDB,-(6)	:PUT @RSDB ON STACK
4268	026652	104404				TYPEO		:TYPE STACK IN OCTAL
4269	026654	032737	000220	022720	10\$:	BIT	#MR,.HLTCT	:TYPTXT MN?
4270	026662	001410				BEQ	11\$	:NO
4271	026664	104402	026670			TYPE	..+2	:.ASCIZ " MR="
4272	026676	017746	152226			MOV	@RSMR,-(6)	:PUT @RSMR ON STACK
4273	026702	104404				TYPEO		:TYPE STACK IN OCTAL
4274	026704	032737	000204	022720	11\$:	BIT	#LA,.HLTCT	:TYPTXT LA?
4275	026712	001410				BEQ	PTDONE	:NO
4276	026714	104402	026720			TYPE	..+2	:.ASCIZ " LA="
4277	026726	017746	152172			MOV	@RSLA,-(6)	:PUT @RSLA ON STACK
4278	026732	104404				TYPEO		:TYPE STACK IN OCTAL
4279	026734	052737	100000	001160	PTDONE:	BIS	#BIT15,ONCE	:SET FORND ERROR FLAG
4280	026742	000207				RTS	PC	
4281								
4282	026744	005037	001176		WAITRY:	CLR	WORK	:CLEAR COUNTER
4283	026750	105777	152130		1\$:	TSTB	@RSCS1	:TEST READY
4284	026754	100406				BMI	2\$	:OK CONT
4285	026756	005237	001176			INC	WORK	:UPDATE COUNTER
4286	026762	005737	001176			TST	WORK	:DONE YET?
4287	026766	001403				BEQ	3\$	:READY DID NOT COME UP
4288	026770	000767				BR	1\$	:CONTINUE WAITING
4289	026772	062716	000002		2\$:	ADD	#2,(SP)	:UPDATE RETURN PC
4290	026776	000207			3\$:	RTS	PC	:RETURN


```
4291 ;RANDOM DATA GENERATOR SUBROUTINE
4292
4293 027000 013737 027142 027146 RANDOM: MOV LONUM,LOSAV
4294 027006 013737 027144 027150 MOV HINUM,HISAV
4295 027014 013700 027142 RAND1: MOV LONUM,R0 ;SET UP R0 WITH 5 DIGITS LOW
4296 027020 013704 027144 MOV HINUM,R4 ;SET UP R1 WITH 5 DIGITS HIGH
4297 027024 012703 000007 MOV #7,R3 ;SET UP SHIFT COUNT
4298 027030 005002 CLR R2 ;CLEAR R2
4299 027032 006300 SHIFT: ASL R0 ;SHIFT R0 LEFT AND
4300 027034 006104 ROL R4 ;ROTATE CARRY INTO LSB OF R1 INTO
4301 027036 006102 ROL R2 ;ROTATE CARRY OUT OF R1 INTO R2
4302 027040 005303 DEC R3 ;DECREMENT R3
4303 027042 001373 BNE SHIFT ;CONTINUE SHIFT LOOP
4304 027044 063700 027142 ADD LONUM,R0 ;ADDN IN NUMBER TO MAKE X 129
4305 027050 005504 ADC R4 ;PROPOGATE CARRY
4306 027052 063704 027144 ADD HINUM,R4 ;ADDN IN NUMBER TO MAKE X 129
4307 027056 005502 ADC R2 ;PROPOGATE CARRY
4308 027060 062700 001057 ADD #1057,R0 ;ADDN LOW CONSTANT
4309 027064 005504 ADC R4 ;PROPOGATE CARRIES
4310 027066 005502 ADC R2 ;PROPOGATE AGAIN
4311 027070 062704 047401 ADD #47401,R4 ;ADDN HIGH CONSTANT
4312 027074 005502 ADC R2 ;PROPOGATE CARRY
4313 027076 062702 000006 ADD #6,R2 ;ADDN HIGHEST CONSTANT
4314 027102 062700 000002 ADD #2,R0 ;REPRIME R0 WITH HIGH DIGIT
4315 027106 005504 ADC R4 ;PROPOGATE CARRY
4316 027110 010037 027142 MOV R0,LONUM ;PUT R0 BACK IN LONUM
4317 027114 010021 MOV R0,(R1)+ ;LOAD WC
4318 027116 005337 001176 DEC WORK
4319 027122 001406 BEQ EXGEN
4320 027124 010437 027144 MOV R4,HINUM ;PUT R1 BACK IN HINUM
4321 027130 010421 MOV R4,(1)+ ;HOLD HINUM FOR PROGRAM
4322 027132 005337 001176 DEC WORK
4323 027136 001326 BNE RAND1
4324 027140 000205 EXGEN: RTS ;RETURN TO PROGRAM
4325 027142 000000 LONUM: 0
4326 027144 000000 HINUM: 0
4327 027146 000000 LOSAV: 0
4328 027150 000000 HISAV: 0
4329 027152 RANEND:
4330
4331 027152 032737 000001 027270 .SUSWR: BIT #BIT0,SWI
4332 027160 001037 BNE XXX
4333 027162 013746 000006 MOV 6,-(SP) ;SAVE 6 ON STACK
4334 027166 013746 000004 MOV 4,-(SP) ;SAVE 4 ON STACK
4335 027172 012737 027212 000004 MOV #1$,4 ;SET UP TRAP ADDRESS
4336 027200 022777 177777 151620 CMP #-1,@SWR ;TEST 177570
4337 027206 001402 BEQ 2$ ;FAKE OUT
4338 027210 000407 BR 3$ ;HARDWARE AVAILABLE
4339 027212 022626 1$: CMP (SP)+,(SP)+ ;ADJUST STACK
4340 027214 012737 000176 001026 2$: MOV #SWREG,SWR ;SET UP SOFTWARE REGISTERS
4341 027222 012737 000174 001030 MOV #DISPREG,DISPLAY
4342 027230 022737 000176 001026 3$: CMP #SWREG,SWR ;1ST TIME THRU?
4343 027236 001004 BNE 4$ ;NO CHANGE STILL 177570
4344 027240 005737 000042 TST 4$ ;ANY XXDP OR ACT
4345 027244 001001 BNE 4$ ;SWR=000000
4346 027246 104422 CNTLU ;GET INITIAL SETTINGS
```



```
4347 027250 012637 000004      4$:  MOV  (SP)+,4      ;REPLACE 4 FROM STACK
4348 027254 012637 000006      MOV  (SP)+,6      ;REPLACE 6 FROM STACK
4349 027260 052737 000001 027270 XXX: BIS  #BIT0,SWI    ;SET THE BEENHEREBIT
4350 027266 000002      RTI                      ;ALL DONE
4351
4352 027270 000000      SWI:  0
4353
4354
4355
4356 027272 005737 000042      .KBDIN: TST  42          ;GOT XXDP OR ACT
4357 027276 001057      BNE OKT          ;YES,GET OUT
4358 027300 022737 000176 001026 CMP  #SWREG,SWR    ;GOT SWITCH-LESS MACHINE?
4359 027306 001053      BNE OKT          ;NO GET OUT
4360 027310 105777 151504      TSTB @TKS          ;HAVE A CHARACTER
4361 027314 100050      BPL OKT          ;NO GET OUT
4362 027316 017737 151500 027440 MOV  @TKB,.MSG
4363 027324 042737 177600 027440 BIC  #177600,.MSG  ;STRIP OFF GARBAGE
4364 027332 122737 000007 027440 CMPB #7,.MSG        ;DO WE HAVE A ^G
4365 027340 001036      BNE OKT          ;NO,GET OUT
4366 027342 104402 027346      TYPE  ..+2        ;.ASCIZ <15><12>'^G'
4367 027354      .CNTLU:
4368 027354 104402 027360      TYPE  ..+2        ;.ASCIZ <15><12>'SWR= '
4369 027370 013746 000176      MOV  SWREG,-(6)    ;PUT SWREG ON STACK
4370 027374 104404      TYPEO          ;TYPE STACK IN OCTAL
4371 027376 104402 027402      TYPE  ..+2        ;.ASCIZ " NEW= "
4372 027414 104410      RDOCT
4373 027416 012637 027440      MOV  (SP)+,.MSG    ;GET NEW VALUE OFF STACK
4374 027422 005737 027442      TST  CTN          ;DID HE TYPE <CR> OF 000000?
4375 027426 001403      BEQ  OKT          ;DONT CHANGE IF <CR>
4376 027430 013737 027440 000176 MOV  .MSG,SWREG    ;CHANGE VALUE OF SWREG
4377 027436 000002      OKT:  RTI          ;ALL DONE-EXIT
4378
4379 027440 000000      .MSG:  0
4380 027442 000000      CTN:  0
```


CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 K 8
TST122 WRITE LOCK TEST PAGE 102

SEQ 0101

4381
4382
4383
4384 027444 000003
4385 027446 000005
4386 027450 000007
4387 027452 000013
4388 027454 000015
4389 027456 000017
4390 027460 000023
4391 027462 000025
4392 027464 000027
4393 027466 000033
4394 027470 000035
4395 027472 000037
4396 027474 000041
4397 027476 000043
4398 027500 000045
4399 027502 000047
4400 027504 000000
4401
4402 027506 000053
4403 027510 000055
4404 027512 000057
4405 027514 000063
4406 027516 000065
4407 027520 000073
4408 027522 000075
4409 027524 000077
4410 027526 000000
4411 027530 000300
4412 030330 000300
4413 031130 000000
4414 031132 000000
4415 000001

;TABLES FOR ILLEGAL FUNCTION TESTS

ILLTAB: 3
5
7
13
15
17
23
25
27
33
35
37
41
43
45
47
0

ILFTB2: 53
55
57
63
65
73
75
77
0

OUTBUF: .BLKW 300
INBUF: .BLKW 300
PARITY: 0 ;USED FOR PARITY TEST 67
CACH: 0 ;USED FOR PARITY TEST 67
.END

2599
3599

1549

1506
1788

1418

CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 105
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0103

CTN 027442
DA = 000004

[illegible]

```

DAO      = 001000
DAOTST   015660
DAOTT    016036
DB       = 000210
DCK      = 100000
DCKTST   014530
DISPLA   001030

```

736#
2932#
2967#
717# 4264
737#
2748#
657# 3705* 3709* 3940* 3947* 3954* 3961* 3968* 3972* 3978* 3985* 3992* 4048*

DISPRE		000174
DKADR		015220
DLT	=	100000
DONCS		012322
DONE		022126
DONEE		002102
DONINT		021124
DRCLR		006040
DRY	=	000200
DS	=	000040

578#	4341											
2842#												
730#	1416											
2345	2365#											
865	3617#											
854	865#											
3454	3458#											
1601#												
731#	2022	2032										
713#	897	1509	1555	1575	1593	1662	1668	1683	1703	1706	1753	1805
1835	1926	1935	1971	1987	2024	2028	2036	2039	2058	2066	2098	2101
2126	2129	2166	2169	2205	2208	2241	2266	2269	2284	2291	2363	2410
2476	2553	2611	2620	2646	2649	2656	2677	2680	2702	2705	2733	2736
2740	2793	2796	2873	2902	2921	2951	2958	2986	2993	3009	3038	3071
3080	3107	3140	3143	3147	3176	3179	3183	3186	3211	3218	3316	3453
3475	3479	3543	3573	3596	3599	4082	4086	4091	4095	4100	4104	4109
4113	4118	4122	4127	4131	4136	4140	4152	4155	4159	4169	4174	4252

DT = 000240
DVNUM 001432
DVTAB 022210
ER = 000002

[illegible]

ERINT	021126
ERR	= 040000
ERRINT	021210
ERRORS	001002
ERRR	012066
ERTST	016202
EXGEN	027140
EXREF	026030
EXTRP	020556
EXTTRP	020564
EXTTST	020042
EXTT1	020364
EXT1	020566
FILCHR	001014
FINDM	025776
FINTST	021232

3465#	3477#		
734#	3728*	3748	
3467	2318#		
647#			
2312			
2999#			
4319	4324#		
4180	4182#		
3295	3371	3377	3382#
3290	3384#		
3287#			
3344#			
3383	3385#		
651#	3653	3654	
1793	4177#		
3476	3481#		

CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 N 8
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0104

FLOTBA 003136
FLOTDA 003436
FLOTWC 003276
FUNDO 016764
HALT 025314
HINUM 027144
HISAV 027150
HLT = 104000

1065#														
1140#														
1103#														
3119#														
4115#														
4294	4296	4306	4320*	4326#										
4294*	4328#													
613#	893	897	901	906	912	918	924	930	936	955	959	964		
970	976	981	986	999	1003	1007	1015	1025	1031	1035	1039	1044		
1052	1056	1060	1071	1082	1090	1094	1098	1109	1119	1127	1131	1135		
1146	1156	1164	1168	1175	1181	1188	1198	1205	1213	1221	1241	1255		
1270	1285	1289	1301	1309	1313	1321	1325	1341	1344	1348	1353	1370		
1375	1380	1395	1403	1406	1418	1426	1450	1458	1474	1477	1480	1483		
1486	1489	1499	1506	1509	1512	1515	1518	1521	1524	1537	1540	1543		
1546	1549	1552	1555	1565	1572	1575	1578	1581	1584	1587	1590	1593		
1613	1616	1619	1624	1627	1630	1633	1647	1650	1654	1656	1659	1662		
1665	1668	1674	1680	1683	1698	1700	1703	1706	1709	1715	1721	1726		
1742	1744	1750	1753	1756	1759	1765	1786	1788	1796	1799	1802	1805		
1811	1817	1830	1832	1835	1838	1841	1847	1853	1859	1874	1876	1882		
1885	1888	1891	1897	1923	1926	1929	1932	1935	1941	1947	1963	1968		
1971	1975	1987	1993	1996	2003	2019	2024	2028	2036	2039	2051	2055		
2058	2061	2064	2066	2095	2098	2101	2107	2113	2116	2119	2122	2126		
2129	2163	2166	2169	2172	2180	2183	2186	2193	2198	2201	2205	2208		
2228	2234	2237	2241	2253	2259	2262	2266	2269	2272	2284	2288	2291		
2303	2320	2337	2360	2363	2407	2410	2466	2472	2476	2482	2489	2492		
2543	2549	2553	2559	2566	2569	2593	2596	2599	2605	2608	2611	2617		
2620	2637	2640	2643	2646	2649	2653	2656	2671	2674	2677	2680	2696		
2699	2702	2705	2727	2730	2733	2736	2740	2782	2790	2793	2796	2802		
2807	2821	2825	2848	2851	2870	2873	2876	2880	2902	2907	2910	2913		
2917	2921	2945	2948	2951	2954	2958	2961	2980	2983	2986	2989	2993		
3006	3009	3013	3017	3024	3029	3035	3038	3041	3044	3047	3050	3053		
3067	3071	3077	3080	3084	3090	3107	3110	3113	3131	3137	3140	3143		
3147	3166	3173	3176	3179	3183	3186	3205	3208	3211	3214	3218	3280		
3316	3320	3332	3336	3342	3355	3358	3370	3374	3380	3402	3408	3427		
3432	3452	3453	3457	3475	3479	3542	3543	3546	3552	3558	3563	3569		
3573	3592	3596	3599	4051	4082	4086	4091	4095	4100	4104	4109	4113		
4118	4122	4127	4131	4136	4140	4152	4155	4159	4169	4174				
650#	3732*	3733*	3734	3735	3749									
2860#	777*	3613*	3683	3698	3700	3702*	3703*	3705	3708*	3709	3715	3744*		
646#														
739#														
2083	2131#													
2146	2210#													
2143	4402#													
2216#														
2142#														
2081	4384#													
2078#														
4412#														
2780#														
3850	3885	3886	3909#											
3428	3433#													
3439#														
3416#														
3419	3431#													
3441	3455#													

HLTADR 001012
IAERR 015274
ICNT 001000
IE = 000100
ILFDN 011046
ILFDNE 011416
ILFTB2 027506
ILF67 011420
ILLFUN 011050
ILLTAB 027444
ILL51 010572
INBUF 030330
INCU 014742
INPUT 023546
INTDON 021006
INTR4 021010
INTR5 020704
INT112 021000
INT114 021112

CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 107
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0105

IR = 000100	725#												
KBDIN = 104416	3680	3724	3930#	3938	3945	3952	3959	3966	3976	3983	3990	3997	4052
KIPAR0= 172340	687#	3297*											
KIPAR1= 172342	688#	3298*											
KIPAR2= 172344	689#	3299*	3345*										
KIPAR7= 172356	690#	3296*											
KIPDR0= 172300	691#	3300*											
KIPDR1= 172302	692#	3301*											
KIPDR2= 172304	693#	3302*											
KIPDR7= 172316	694#	3303*											
LA = 000204	716#	2303	2320	4274									
LAD 001010	649#	778*	1231*	1245*	1259*	3704*	3710	3712	3715				
LASTSC 017332	3192#												
LATDON 012132	2317	2321#											
LATST 012010	2301#												
LBT = 002000	733#												
LONUM 027142	4293	4295	4304	4316*	4325#								
LOP1 004056	1231	1232#											
LOP2 004126	1245	1246#											
LOP3 004176	1259	1260#											
LOSAV 027146	4293*	4327#											
MAXREF 026062	4177	4188#											
MAXRF1 026064	4186	4189#											
MEMOUT 020572	3386#												
MODDON 022076	3610#												
MODNUM 021234	3491#												
MPRO = 172100	751#	2380											
MR = 000220	718#	924	981	1198	1205	1213	1221	1624	2019	2337	2363	4269	
MULTII 001402	799#												
N = 000123	612#	794	798#	874	878#	937	941#	987	991#	1008	1009#	1017	1021#
	1046	1047#	1062	1063#	1074	1075#	1083	1084#	1099	1100#	1113	1114#	1120
	1121#	1136	1137#	1150	1151#	1157	1158#	1170	1171#	1176	1177#	1182	1183#
	1189	1190#	1199	1200#	1206	1207#	1214	1215#	1222	1226#	1275	1279#	1291
	1292#	1303	1304#	1315	1316#	1327	1331#	1354	1358#	1381	1382#	1427	1428#
	1462	1466#	1490	1494#	1525	1529#	1556	1560#	1594	1598#	1635	1639#	1686
	1690#	1727	1731#	1773	1777#	1818	1822#	1860	1864#	1902	1906#	1952	1956#
	1976	1980#	2005	2009#	2068	2072#	2132	2136#	2211	2215#	2274	2278#	2296
	2300#	2304	2305#	2322	2326#	2366	2370#	2428	2432#	2501	2505#	2579	2583#
	2621	2625#	2658	2662#	2683	2687#	2707	2711#	2741	2745#	2837	2841#	2853
	2857#	2884	2888#	2927	2931#	2962	2966#	2994	2998#	3054	3058#	3091	3095#
	3115	3119#	3148	3152#	3187	3191#	3223	3227#	3282	3286#	3388	3392#	3411
	3415#	3434	3438#	3460	3464#	3482	3486#	4056	4060#				
NED = 010000	728#	2915	2919										
NEDDON 015566	2896	2923#											
NEDTST 015410	2889#												
NNDD 015650	2922	2924	2926#										
NNOOP 005330	1495#												
NNOP21 005646	1561#												
NOOP 005204	1467#												
NOOP21 005510	1530#												
NOPAR 012606	2414	2421	2425#										
NOWGO 002106	847	864	870#										
NXM 007612	1898#	4201											
NXMTSM 010144	1981#												
OBUF SV 001102	661#	780*	870*	1469	1484	1501	1516	1532	1547	1567	1582	1642	1648
	1676	1693	1717	1737	1761	1779	1781	1813	1825	1849	1854	1867	1869

RANDOM	027000
RAND1	027014
RANEND	027152
RANNU	001150
RANTS	004026
RDLIN =	104412
RDOCT =	104410
RDTST	006462
RDTSTB	007240
RMRT1	013762
RMRT2	014140
RMRT3	014250
RMRT4	014360
RSAS	001122

[illegible]

CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 109
CROSS REFERENCE TABLE -- USER SYMBOLS

D 9

SEQ 0107

RSBA	001112	668#	881*	904	945*	962	1049*	1050	1053*	1054	1057*	1058	1067*	1068
		1078*	1079*	1080	1263	1268	1317*	1319	1322*	1323	1469*	1484	1501*	1516
		1532*	1547	1567*	1582	1642*	1648	1675	1693*	1716	1737*	1760	1781*	1812
		1825*	1848	1869*	1892	1913*	1942	1958*	1965	1984*	2012*	2059	2084*	2114
		2147*	2181	2196	2220*	2224	2245*	2250	2397*	2439*	2455*	2477	2516*	2532*
		2554	2586*	2627*	2664*	2689*	2713*	2759*	2766*	2783*	2818	2827*	2844*	2862*
		2942*	2977*	3019*	3045	3099*	3108	3122*	3156*	3202*	3229*	3244*	3262*	3308*
		3324*	3348*	3364*	3424*	3447*	3472*	3521*	3553	3946*	3947	4000*	4040*	4045*
		4071*	4142*	4161*	4240									
RSBAB	001146	682#	1318*											
RSCS1	001104	665#	880*	899	944*	957	996*	997	1000*	1001	1004*	1005	1011*	1012*
		1013	1281*	1283	1286*	1287	1342	1345*	1346	1471*	1472	1503*	1504	1534*
		1535	1553	1569*	1570	1591	1606*	1625	1644*	1651*	1652	1660	1695*	1696
		1704	1739*	1740	1751	1783*	1784	1797	1827*	1828	1836	1871*	1872	1883
		1924	1961*	1985*	1994	1997*	2021*	2037	2086*	2089	2092	2127	2154*	2157
		2160	2203	2221*	2222	2246*	2247	2263*	2264	2399*	2400	2408	2441*	2442
		2458*	2459	2474	2490	2518*	2519	2535*	2536	2551	2567	2588*	2589	2591*
		2594	2612*	2618	2634*	2647	2654	2666*	2667	2678	2691*	2692	2694*	2703
		2720*	2734	2738	2761*	2762	2773*	2780	2785*	2786	2794	2828*	2829	2846*
		2864*	2865	2874	2908	2943*	2952	2978*	2987	3007	3015	3021*	3022	3025
		3027	3078	3082	3102*	3103	3105	3124*	3127	3129	3145	3158*	3159	3162
		3164	3181	3203*	3212	3239*	3240	3247*	3248	3272*	3273	3309*	3312	3314
		3318	3326*	3328	3330	3334	3350*	3351	3353	3356	3365*	3366	3368	3372
		3398*	3406	3425*	3448*	3455	3473*	3477	3525*	3531*	3537	3574*	3579*	3597
		3603*	4001*	4002	4004	4041*	4042	4046*	4048	4049	4074*	4075	4145*	4146
		4148	4164*	4165	4220	4283								
RSCS1B	001140	679#	1282*	2914*										
RSCS2	001106	666#	800*	804*	815*	879*	886*	891	894*	943*	950*	952*	953	956*
		1023	1026*	1027	1029	1032*	1033	1036*	1037	1040*	1041*	1042	1045*	1293*
		1294*	1298	1336	1349	1365	1392	1404	1416	1607	1671	1712	1747	1791
		1808	1844	1879	1938	1960*	1973	1982*	1990	1998	2102	2175	2188	2231
		2256	2402	2461	2486	2538	2563	2602	2614	2760*	2778*	2779*	2797	2898*
		2903	2915	2919	3030	3060*	3062	3085	3097*	3101*	3120*	3126*	3132	3154*
		3161*	3167	3238*	3246*	3250*	3277	3528*	3530*	3536*	3549	3593*	3977*	3978
		4033*	4204*	4205*	4228									
RSCS2B	001142	680#	1295*											
RSDA	001114	669#	882*	910	946*	968	1124*	1125	1128*	1129	1132*	1133	1142*	1143
		1152*	1153*	1154	1249	1253	1481	1513	1544	1579	1603*	1614	1657	1701
		1754	1794	1833	1886	1927	2011*	2020*	2120	2170	2341*	2343*	2344	2356
		2457*	2534*	2635*	2641	2764*	2845*	2849	2863*	2899*	2918	2941*	2976*	3042
		3193*	3242*	3261*	3323*	3363*	3470*	3523*	3561	3567	3571	3953*	3954	4036*
		4072*	4079*	4083*	4092*	4101*	4110*	4119*	4128*	4137*	4143*	4162*	4245	
RSDB	001126	674#	948*	1335	1360*	1361*	1372	1377	1387*	1400	1412*	1422	1433*	1439*
		1447	1455	4267										
RSDS	001116	670#	895	1507	1573	1666	1803	1933	2022	2026	2032	2034	2056	2099
		2167	2282	2289	2644	2675	2700	2723	2731	2791	2871	2949	2956	2984
		2991	3036	3138	3174	3209	3216	3594	3960*	3961	4080	4084	4089	4093
		4098	4102	4107	4111	4116	4120	4125	4129	4134	4138	4150	4157	4172
		4255												
RSDT	001132	676#	816	820	824	828	830	4262						
RSER	001120	671#	801*	883*	916	947*	974	1161*	1162	1165*	1166	1172*	1173	1178*
		1179	1184*	1185*	1186	1475	1496*	1522	1538	1562*	1588	1604*	1617	1681
		1969	2096	2124	2164	2206	2235	2239	2260	2267	2281*	2286	2609	2638
		2651	2669*	2672	2697	2728	2788	2823	2868	2878	2900	2946	2959	2981
		3000*	3051	3069	3141	3177	3184	3206	3967*	3968	4153	4224		
RSLA	001124	673#	2301	2309	2313	2350	3972	4277						

CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 110
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0108

RSMR	001130	675#	884*	922	951*	979	1193*	1194	1201*	1202*	1203	1208*	1209	1216*
		1217	1605*	1620	2015	2331	2335	2348	3061	3068*	3991*	3992	4272	
RSMRR	024056	602	3989#											
RSREG	026232	3738	4209#											
RSVCPS	001136	678#	3396*	3420*	3442*	3469*	3519*	3608	3609*					
RSVEC	001134	677#	3395*	3419*	3441*	3467*	3518*	3577*	3608*					
RSWC	001110	667#	885*	928	949*	984	1087*	1088	1091*	1092	1095*	1096	1105*	1106
		1115*	1116*	1117	1235	1239	1305*	1307	1310*	1311	1470*	1478	1502*	1510
		1533*	1541	1568*	1576	1602*	1631	1643*	1645	1663	1694*	1707	1738*	1757
		1782*	1800	1826*	1839	1870*	1889	1914*	1930	1959*	1983*	2013*	2062	2085*
		2117	2148*	2184	2199	2217*	2242*	2398*	2440*	2456*	2517*	2533*	2587*	2606
		2628*	2665*	2690*	2714*	2757*	2767*	2784*	2804	2826*	2843*	2861*	2935*	2937*
		2940*	2970*	2972*	2975*	3020*	3039	3100*	3111	3123*	3157*	3196*	3198*	3201*
		3232*	3234*	3237*	3245*	3265*	3269*	3271*	3307*	3325*	3349*	3362*	3423*	3446*
		3471*	3522*	3544	3939*	3940	3999*	4037*	4073*	4144*	4163*	4250		
RSWCB	001144	681#	1306*											
RSWCWT	006712	1743#												
RS04DT	001162	747#	818*	822*	826*	834*	2749	2754	2809	2813	2933	2938	2968	2973
		3194	3199	3230	3235	3251	3263	3267	3559	3565				
RW	= 000006	695#	3300	3301	3302	3303								
SAVBAD	001100	660#												
SAVEE	001172	753#	3516*	3600*										
SC	= 100000	724#												
SCOPE	= 104400	797	877	940	990	1008	1020	1046	1062	1074	1083	1099	1113	1120
		1136	1150	1157	1170	1176	1182	1189	1199	1206	1214	1225	1278	1291
		1303	1315	1330	1357	1381	1427	1465	1493	1528	1559	1597	1638	1689
		1730	1776	1821	1863	1905	1955	1979	2008	2071	2135	2214	2277	2299
		2304	2325	2369	2431	2504	2582	2624	2661	2686	2710	2744	2840	2856
		2887	2930	2965	2997	3057	3094	3118	3151	3190	3226	3285	3391	3414
		3437	3463	3485	3610	3617	3923#	4059	4199	4200				
SECT	017476	3227#												
SEEC	026566	4231	4258#											
SETPAT	016650	3096#												
SHIFT	027032	4299#	4303											
SILO	004716	1384#												
SILOB	004506	1334#												
SILOFL	005070	1430#												
SRAS	024032	601	3982#											
SRCS2	024006	600	3975#											
SRSBA	023656	595	3944#											
SRSCS1	024102	603	3996#											
SRSDA	023702	596	3951#											
SRSDS	023726	597	3958#											
SRSER	023752	598	3965#											
SRSLA	023776	599	3972#	3973										
SRSWC	023632	594	3937#											
SRO	= 177572	686#	2384*	3292	3304*	3382*								
STRT	000230	582	588#											
STTEST	001516	810	813#	851										
SUSWR	= 104420	784	3931#	4030										
SWI	027270	4029*	4331	4349*	4352#									
SWR	001026	656#	809	835	857	1789	3620	3681	3683	3685	3696	3725	3729	3739
		3742	3939	3946	3953	3960	3967	3977	3984	3991	4001	4031	4034	4038
		4053	4063	4068	4179	4192	4336	4340*	4342	4358				
SWRCHG	024464	4044	4048#											
SWREG	000176	579#	4063	4340	4342	4358	4369	4376*						

[illegible]

CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 G 9
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0110

TST21	003554	1170#
TST22	003576	1176#
TST23	003620	1182#
TST24	003644	1189#
TST25	003702	1199#
TST26	003730	1206#
TST27	003766	1214#
TST3	002360	940#
TST30	004024	1225#
TST31	004244	1278#
TST32	004316	1291#
TST33	004362	1303#
TST34	004432	1315#
TST35	004504	1330#
TST36	004606	1357#
TST37	004714	1381#
TST4	002572	990#
TST40	005066	1427#
TST41	005202	1465#
TST42	005326	1493#
TST43	005506	1528#
TST44	005644	1559#
TST45	006036	1597#
TST46	006232	1638#
TST47	006460	1689#
TST5	002664	1008#
TST50	006646	1730#
TST51	007020	1776#
TST52	007236	1821#
TST53	007432	1863#
TST54	007626	1905#
TST55	010036	1955#
TST56	010142	1979#
TST57	010270	2008#
TST6	002712	1020#
TST60	010570	2071#
TST61	011046	2135#
TST62	011416	2214#
TST63	011730	2277#
TST64	012006	2299#
TST65	012022	2304#
TST66	012132	2325#
TST67	012322	2369#
TST7	003052	1046#
TST70	012624	2431#
TST71	013166	2504#
TST72	013560	2582#
TST73	013760	2624#
TST74	014136	2661#
TST75	014246	2686#
TST76	014356	2710#
TST77	014526	2744#
TTAGG	002362	942#
TTMOUT	010566	2031
TTVEC	021734	3577
TYPE =	104402	790

2066#
3583#
811

839

845

861

862

2319

2925

3622

3727

3731

3737

3789

CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 PAGE 113
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0111

TYPEO = 104404	3829	3875	3894	3899	3902	3905	3924#	4019	4022	4026	4062	4077	4087
TYPES = 104406	4096	4105	4114	4123	4132	4170	4196	4211	4214	4219	4223	4227	4234
UNCMP 001156	4239	4244	4249	4254	4261	4266	4271	4276	4366	4368	4371	4256	4263
UNITST 007630	3736	3925#	4213	4216	4221	4225	4229	4236	4241	4246	4251		
UNITSV 001154	4268	4273	4278	4370									
UNNUM 001152	843	860	3926#										
	745#	807*	813	832	848*	855*	1487	1497	1519	1550	1563	1585	1628
	2108	3002	3010	3048	3073								
	1907#												
	744#	806*	813	832*	2892	3499							
	743#	808*	815	842	850*	856*	859	879	894	943	956	1045	1293
	1296	1338	1364	1391	1609	1669	1710	1745	1806	1842	1877	1909	1916
	1936	1988	1999	2040	2103	2176	2189	2229	2254	2404	2462	2484	2539
	2561	2600	2779	2798	3032	3064	3086	3134	3169	3275	3501	3509	3536
UP = 000000	3547	4021*	4068*	4205									
WAITRY 026744	696#	3300	3301	3302	3303								
	1655	1699	1743	1787	1831	1875	1922	1962	1986	2592	2636	2670	2695
WC = 000010	2847	2944	2979	3204	3426	3474	4282#						
	711#	930	986	1090	1094	1098	1119	1309	1313	1480	1512	1543	1578
	1633	1647	1665	1709	1759	1802	1841	1891	1932	2064	2119	2186	2201
	2608	3041	3113	3546	4051	4247							
WCE = 040000	729#												
WCEDNB 013522	2531	2573#											
WCEDNE 013136	2454	2496#											
WCEDON 013160	2449	2500#											
WCEDOS 013544	2512	2526	2577#										
WCETB 013220	2516#	2576											
WCETSB 013170	2510#												
WCETST 012626	2437#												
WCETT 012634	2439#	2499											
WORK 001176	766#	1194*	1195*	1196	1209*	1210*	1211	1217*	1218*	1219	1227*	1362*	1368*
	1620*	1621*	1622	1779*	1780*	1824*	1854*	1855	1867*	1868*	2014*	2017*	2040*
	2044	2046*	2144*	2153*	2173	2194*	2195*	2196	2218*	2219*	2220	2243*	2244*
	2245	2308*	2311*	2342*	2346*	2722*	2725*	2890*	2892	2895*	2918*	3253*	3255*
	3259*	3399*	3400*	3449*	3450*	3498*	3499	3505*	3585*	3586	3590*	3602	4024*
	4025*	4037	4282*	4285*	4286	4318*	4322*						
WORK1 001200	767#	2149*	2150*	2151	2352*	2354*	3494*	3507	3513*	3575	4034*	4035*	4036
WORK2 001202	768#	2751*	2753*	2756*	2757	2784	4031*	4053					
WRCKT 006650	1735#												
WRCKTB 007434	1866#												
WRTLCK 024520	585	4061#											
WRTST 006234	1640#												
WRTSTB 007022	1778#	4187											
WT 024436	4042#	4047											
WTDV 021502	3535#												
WTDV1 022052	3576	3601	3606#										
WTDV2 021732	3578	3581#	3604										
WTE 024446	4039	4045#											
WVP = 000004	750#	2382											
XXX 027260	4332	4349#											
ZERONE 004610	1359#												
SENDAD 022172	573	3626#											
SEND1 022202	3624	3628#											
. = 031134	571#	572#	574#	577#	581#	584#	587#	593#	605#	644#	659#	782	790
	811	839	862	863#	892	896	900	905	911	917	923	929	935
	954	958	963	969	975	980	985	998	1002	1006	1014	1024	1034

[illegible]

CZRSBHO RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

J 9
MACY11 30A(1052) 13-OCT-80 15:16 PAGE 115
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0113

		3623	3680	3724	3729	3742
.TBIT	022206	3630#				
.TRAP	023566	775	3917#			
.TRP	023604	3920	3921#			
.TYPE	022230	3643#	3924			
.TYPEB	022722	3755#				
.TYPEO	022732	3757#	3925			
.TYPES	022742	3759#	3926			

END	1#	555#													
LDPDR	698#	3300	3301	3302	3303										
NEWST	1#	555#	794	874	937	987	1008	1017	1046	1062	1074	1083	1099	1113	1120
	1136	1150	1157	1170	1176	1182	1189	1199	1206	1214	1222	1275	1291	1303	1315
	1327	1354	1381	1427	1462	1490	1525	1556	1594	1635	1686	1727	1773	1818	1860
	1902	1952	1976	2005	2068	2132	2211	2274	2296	2304	2322	2366	2428	2501	2579
	2621	2658	2683	2707	2741	2837	2853	2884	2927	2962	2994	3054	3091	3115	3148
	3187	3223	3282	3388	3411	3434	3460	3482	4056						
ODT11	1#														
ODT11X	1#														
POP	1#	641#	3691	3694	3790	3821	3869								
PRINT	1#	641#	790	811	839	862	2319	2925	3622	3731	3737	3829	3874	3894	3898
	4019	4022	4026	4062	4077	4087	4096	4105	4114	4123	4132	4170	4196	4211	4214
	4219	4223	4227	4234	4239	4244	4249	4254	4261	4266	4271	4276	4366	4368	4371
PUSH	1#	641#	3688	3761	3807	3844									
SCOPE.	1#	794#	874#	937#	987#	1008#	1017#	1046#	1062#	1074#	1083#	1099#	1113#	1120#	1136#
	1150#	1157#	1170#	1176#	1182#	1189#	1199#	1206#	1214#	1222#	1275#	1291#	1303#	1315#	1327#
	1354#	1381#	1427#	1462#	1490#	1525#	1556#	1594#	1635#	1686#	1727#	1773#	1818#	1860#	1902#
	1952#	1976#	2005#	2068#	2132#	2211#	2274#	2296#	2304#	2322#	2366#	2428#	2501#	2579#	2621#
	2658#	2683#	2707#	2741#	2837#	2853#	2884#	2927#	2962#	2994#	3054#	3091#	3115#	3148#	3187#
	3223#	3282#	3388#	3411#	3434#	3460#	3482#	4056#							
SET	3910#	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932				
TYPADR	1#														
TYPBIT	1#	641#													
TYPCHR	1#	641#													
TYPDEC	1#	641#													
TYPLIN	1#	641#													
TYPOCS	1#	641#	841	859											
TYPOCT	1#	641#	3735	4212	4215	4220	4224	4228	4235	4240	4245	4250	4255	4262	4267
	4272	4277	4369												
TYPTXT	1#	641#	790	811	839	862	2318	2925	4019	4022	4026	4061	4077	4087	4096
	4105	4114	4123	4132	4170	4196	4211	4214	4218	4222	4226	4234	4239	4244	4249
	4254	4261	4266	4271	4276	4366	4367	4371							
\$CATCH	1#	555#	571												
\$CMTAG	1#	555#	644												
\$DONE	1#	555#	3615												
\$EQUAT	1#	555#	612												
\$HLT	1#	555#	3716												
\$KBDIN	1#	555#	4355												
\$KMMR	1#														
\$OCTAL	1#	555#	3749												
\$POWER	1#	555#	3797												
\$RAND2	1#														
\$RAND4	1#														
\$RDDEC	1#														
\$RDLIN	1#	555#	3877												
\$RDOCT	1#	555#	3837												
\$SCOPE	1#	555#	3673												
\$SET	3910#	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932				
\$SETUP	1#	555#	770												
\$SMMR	1#														
\$SUSWR	1#	555#	4330												
\$SWDOC	1#	555#	559												
\$TRAP	1#	555#	3910												
\$TYPE	1#	555#	3633												
\$TYPEA	1#														

CZRSBH0 RH11-RS03/04 BFT
CZRSBH.P11 13-OCT-80 15:14

MACY11 30A(1052) 13-OCT-80 15:16 L 9
PAGE 118
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0115

\$TYPED 1#
\$UMMR 1#

. ABS. 031134 000

ERRORS DETECTED: 0

CZRSBH.BIN,CZRSBH.LST/CRF/SOL/NL:TOC=CZRSBH.SML,CZRSBH.P11
RUN-TIME: 29 46 4 SECONDS
RUN-TIME RATIO: 178/81=2.1
CORE USED: 22K (43 PAGES)